

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Metody detekce hran v biomedicínských obrazech

Methods of Edge Detection in Biomedical Images

Zadání bakalářské práce

Student:

Kristýna Kubíková

Studijní program:

B2649 Elektrotechnika

Studijní obor:

3901R039 Biomedicínský technik

Téma:

Metody detekce hran v biomedicínských obrazech
Methods of Edge Detection in Biomedical Images

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Nastudování základních principů zpracování digitálního obrazu.
2. Nastudování základních principů segmentace obrazu.
3. Rešerše matematických algoritmů pro detekci hran v obraze.
4. Rešerše aplikací detektorů hran pro klinická obrazová data.
5. Tvorba datové báze vybraných klinických obrazů z variantních obrázkových modalit.
6. Design a realizace vybraných konvenčních metod pro detekci obrazových hran.
7. Design a realizace vybraných nekonvenčních metod pro detekci obrazových hran.
8. Testování navržených algoritmů pro vybrané klinické nálezy.
9. Evaluace a kvantitativní testování navržených detektorů hran.
10. Design graficko-uživatelského rozhraní pro edukativní účely detekce hran v biomedicínských obrazech.
11. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] DHAWAN, Atam P. *Medical image analysis*. Piscataway, N.J.: IEEE Press, c2003. ISBN 0-471-45131-2.
- [2] RANGAYYAN, Rangaraj M. *Biomedical signal analysis*. Second edition. Hoboken, New Jersey: John Wiley & Sons, [2015]. ISBN 978-0-470-91139-6.
- [3] SURI, Jasjit S, David L WILSON a Swamy LAXMINARAYAN. *Handbook of biomedical image analysis*. New York: Kluwer Academic/Plenum Publishers, c2005. ISBN 0-306-48605-9.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Kubiček, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



A handwritten signature in blue ink, consisting of a stylized 'J' and 'K'.

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

A handwritten signature in blue ink, consisting of a stylized 'P' and 'B'.

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární
prameny a publikace, ze kterých jsem čerpala.

V Ostravě 30. dubna 2019

Mulíkova
.....

Chtěla bych poděkovat panu Ing. Janu Kubíčkovi, Ph.D. za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnoval.

Abstrakt

Bakalářská práce popisuje zpracování obrazu, vlastnosti obrazu, generátory šumů a hranové detektory. Práce se věnuje metodám detekce hran v biomedicínských obrazech s použitím reálných datových sad. Cílem této bakalářské práce jsou experimenty spočívající v porovnání odolnosti hranových detektorů vůči šumu. Dalším cílem je vlastní implementace hranových operátorů a aplikace s grafickým uživatelským rozhraním pro edukativní účely. Všechny tyto experimenty byly implementovány v prostředí MATLAB.

Klíčová slova: zpracování obrazu, detekce hran, hranové detektory

Abstract

This bachelor thesis describes image processing, image attributes, noise generators and edge detectors. This thesis is dedicated to the methods of edge detection in biomedical images using real datasets. The aim of this bachelor thesis are experiments providing information about the detector noise resistance. Another aim is own implementation of selected edge detection operators and an application with a graphical user interface for educational purpose. All these experiments were implemented in MATLAB environment.

Key Words: image processing, edge detection, edge detectors

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	13
1 Úvod	14
2 Úvod do zpracování obrazu	15
2.1 Proces zpracování obrazu	15
2.2 Formát dat	18
3 Obecná řešerše matematických metod pro hranové operátory	21
3.1 Konvoluce	21
3.2 Metody detekce hran založené na první derivaci	22
3.3 Metody detekce hran založené na druhé derivaci	25
3.4 Detekce hran využívající fuzzy logiku	28
4 Biomedicínské aplikace detektorů hran	29
5 Cíle modelování detekce hran	31
6 Testování detektorů hran na reálných medicínských datech	32
6.1 Metoda Sobel	32
6.2 Metoda Roberts	34
6.3 Metoda Prewitt	35
6.4 Metoda Canny	36
6.5 Metoda Laplacian	37
6.6 Metoda Zero-cross	38
7 Šumové generátory	39
7.1 Gaussovský šum	39
7.2 Salt & Pepper	40
7.3 Speckle	41
7.4 Localvar	42
8 Design experimentu	44
8.1 Popis datových sad	44
8.2 Konverze obrazu	46
8.3 Průběh jednoho experimentu	46

8.4	Aplikace pro vykonání experimentů	47
8.5	Vlastní implementace hranových oprátorů	49
8.6	Evaluační techniky	51
9	Výsledky experimentů	52
9.1	Porovnání změny chování operátoru při změně datasetu	52
9.2	Srovnání vlastní a nativní implementace	53
9.3	Vliv jednotlivých šumů na metody detekce hran	54
9.4	Porovnání jednotlivých metod	55
9.5	Porovnání průměrných výsledků jednotlivých metod	57
9.6	Porovnání výsledků klasické metody Sobel a metody Sobel s použitím fuzzy logiky	58
10	Graficko-uživatelské rozhraní	60
11	Závěr	62
	Literatura	63
	Seznam příloh	66

Seznam použitých zkratek a symbolů

3D	– trojrozměrný
BMP	– Microsoft Windows bitmap
CMY	– Cyan, Magenta, Yellow
CMYK	– Cyan, Magenta, Yellow, Black
CT	– výpočetní tomografie
DICOM	– Digital Imaging and Communications in Medicine
GIF	– Graphics Interchange Format
GUI	– grafické uživatelské prostředí
JPEG	– Joint Photographic Group
LoG	– Laplacián Gaussiánu
MB	– megabyte
MRI	– magnetická rezonance
MSE	– střední kvadratická chyba
PET	– Pozitronová emisní tomografie
PNG	– Portable Network Graphics
RGB	– Red, Green, Blue
corr	– korelace
∇^2	– Laplaceův operátor

Seznam obrázků

1	Vzorkování, vlevo rozlišení 256x256, vpravo 56x56[4]	16
2	Kvantování, vlevo 256 jasových úrovní, vpravo 4 jasové úrovně[4]	16
3	Segmentace práhováním[6]	17
4	Segmentace obrazu pomocí narůstání oblastí[7]	18
5	Modely hran[4]	21
6	Hodnoty jasu obrazu, Konvoluční maska, Výsledný pixel[12]	22
7	Směr hrany je kolmý na směr gradientu [4]	22
8	Frůběh jasové funkce a její první a druhá derivace [15]	25
9	Příčný řez operátoru LoG	27
10	Diagram znázorňující implementaci operátoru využívajícího fuzzy logiku	28
11	Panoramatický rentgenový snímek[21]	29
12	Výsledek detekce hran[21]	29
13	Původní snímek z MRI[22]	30
14	Hrany tumoru [22]	30
15	Původní obraz s vyznačenými hranami tumoru[22]	30
16	Detekce hran s použitím Sobelova operátoru na nezašuměném obraze s práhem a) 0.025, b) 0.05, c) 0.1 a na obraze zašuměném šumem Salt & Pepper s hustotou šumu d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašuměném obraze byl použit práh 0.05	33
17	Detekce hran s použitím operátoru Roberts na nezašuměném obraze s práhem a) 0.025, b) 0.05, c) 0.1 a na obraze zašuměném šumem Salt & Pepper s hustotou šumu d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašuměném obraze byl použit práh 0.05	34
18	Detekce hran s použitím operátoru Prewitt na nezašuměném obraze s práhem a) 0.025, b) 0.05, c) 0.1 a na obraze zašuměném šumem Salt & Pepper s hustotou šumu d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašuměném obraze byl použit práh 0.05	35
19	Detekce hran s použitím operátoru Canny na nezašuměném obraze s práhem a) 0.025, b) 0.05, c) 0.1 a na obraze zašuměném šumem Speckle s rozptylem šumu d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašuměném obraze byl použit práh 0.05	36
20	Detekce hran s použitím operátoru Laplacián na nezašuměném obraze s práhem a) 0.0025, b) 0.005, c) 0.01 a na obraze zašuměném Gaussovským šumem s rozptylem d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašuměném obraze byl použit práh 0.005	37
21	Detekce hran s použitím operátoru Zero-cross na nezašuměném obraze s práhem a) 0.0025, b) 0.005, c) 0.01 a na obraze zašuměném šumem Localvar s rozptylem d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašuměném obraze byl použit práh 0.005	38

22	Gaussovský šum s parametrem rozptylu s hodnotou a) původní obraz ,b) 0.01, c) 0.025, d) 0.05, e) 0.1, f) 0.25	40
23	Šum Salt & Pepper s parametrem hustoty šumu s hodnotou a) původní obraz ,b) 0.01, c) 0.025, d) 0.05, e) 0.1, f) 0.25	41
24	Šum Speckle s parametrem rozptylu s hodnotou a) původní obraz ,b) 0.01, c) 0.025, d) 0.05, e) 0.1, f) 0.25	42
25	Gaussovský šum s parametrem rozptylu s hodnotou a) původní obraz ,b) 0.01, c) 0.025, d) 0.05, e) 0.1, f) 0.25	43
26	Ukázka datových sad a)-c) CT - kalcifikace cév , d)-f) CT- játra, g)-i) MRI - kloubní chrpavka, j)-l) MRI - krevní cévy	45
27	Vývojový diagram jendoho experimentu	46
28	Detekce hran na a) nezašuměném a b) zašuměném obraze s Gaussovským šumem	46
29	Vývojový diagram aplikace pro provedení experimentů	47
30	Vývojový diagram aplikace pro provedení experimentů	47
31	Vývojový diagram aplikace pro provedení experimentů	48
32	Vývojový diagram pro vlastní konvoluci	49
33	Vývojový diagram pro implementaci hranového operátoru	50
34	Vývoj střední kvadratické chyby při změně datasetu pro metodu Sobel s práhem 0.1 zašumělé šumem a)Gaussovským, b)Salt & Pepper, c)Speckle, d)Localvar . .	52
35	Vývoj korelace při změně datasetu pro metodu Laplacián s práhem 0.01zašumělé šumem a)Gaussovským, b)Salt & Pepper, c)Speckle, d)Localvar	53
36	Porovnání vývoje korelace po detekci hran s použitím nativní funkce MATLABu a vlastní implmelentací funkce a to jak s použitím nativní funkce pro konvoluci ,tak s vlastní implementací konvoluce pro metodu Sobel použitou na obrazy zašumělé a) Gaussovským šumem, b) šumem Salt & Pepper a pro metodu Roberts na obrazy zašumělé c)Gaussovským šumem, d) šumem Salt & Pepper	54
37	Změna střední kvadratické chyby při změně šumu pro metodu a) Sobel, b) Canny	55
38	Změna korelace při změně šumu pro metodu a) Sobel, b) Canny	55
39	Porovnání výsledků korelace pro jednotlivé metody pro a), c) šum Salt&Pepper a b), d) šum Speckle s práhy a),b) 0.1, c),d) 0.01	56
40	Porovnání výsledků střední kvadratické chyby pro jednotlivé metody pro a), c) Gaussovský šum a b), d) šum Speckle s práhy a),b) 0.1, c),d) 0.01	57
41	Porovnání výsledků střední kvadratické chyby a), b) a korelace c), d) pro metody Sobel a Sobel s použitím fuzzy logiky pro a), c)Gaussovský šum a b), d) šum Salt&Pepper.	59
42	Vývojový diagram pro práci s GUI	60
43	Vstupní okno uživatelského prostředí	60
44	Okno detekce hran	61
45	Ukázka obrazů první datové sady CT obsahující kalcifikaci cév	67

46	Ukázka obrazů druhé datové sady CT jater	68
47	Ukázka obrazů třetí datové sady MRI kloubní chrupavky	69
48	Ukázka obrazů čtvrté datové sady MRI krevních cév	70
49	Gaussovský šum s rozptylem a)-c) 0.025, d)-f) 0.1, g)-i) 0.25	71
50	Šum Salt & Pepper s hustotou a)-c) 0.025, d)-f) 0.1, g)-i) 0.25	72
51	Šum Speckle s rozptylem a)-c) 0.025, d)-f) 0.1, g)-i) 0.25	73
52	Šum Speckle s rozptylem a)-c) 0.025, d)-f) 0.1, g)-i) 0.25	74
53	a) Sobel s práhem 0.05 b) Sobel s práhem 0.05 na obraze zašuměném šumem Salt & Pepper s hustotou 0.05	75
54	a) Laplacian s práhem 0.0075 b) Laplacian s práhem 0.0075 na obraze zašuměném šumem Speckle s rozptylem 0.1	75
55	a) Canny s práhem 0.25 b) Canny s práhem 0.25 na obraze zašuměném šumem Localvar s rozptylem 0.05	76
56	a) Canny s práhem 0.25 b) Canny s práhem 0.25 na obraze zašuměném šumem Salt & Pepper s hustotou 0.05	76
57	a) Prewitt s práhem 0.05 b) Prewitt s práhem 0.05 na obraze zašuměném šumem Speckle s rozptylem 0.25	77
58	a) Roberts s práhem 0.05 b) Roberts s práhem 0.05 na obraze zašuměném šumem Salt & Pepper s hustotou 0.025	77
59	a) ZeroCross s práhem 0.0075 b) ZeroCross s práhem 0.05 na obraze zašuměném Gaussovským šumem s rozptylem 0.025	78
60	Porovnání výsledků střední kvadratické chyby a), b) a korelace c), d) pro metody Sobel a Sobel s použitím fuzzy logiky pro a), c) šum Speckle a b), d) šum Localvar.	79
61	Porovnání výsledků střední kvadratické chyby pro jednotlivé metody pro a), c) Salt & Pepper a b), d) šum Localvar.	80
62	Porovnání výsledků korelace pro jednotlivé metody pro a), c) Salt & Pepper a b), d) šum Localvar.	81
63	Změna korelace při změně šumu pro metodu a) Prewitt, b) ZeroCross	82
64	Změna střední kvadratické chyby při změně šumu pro metodu a) Prewitt, b) ZeroCross	82

Seznam tabulek

1	Ukázka výsledků experimetu	49
2	Tabulka průměrných hodnot korelace	57

1 Úvod

Zrak patří mezi základní smysly člověka a je důležitým nástrojem pro vnímání okolního světa. Pomocí zraku náš mozek přijímá velké množství informací. Lidé se vždy snažili o uchování této informace do podoby, která je přenositelná a může sloužit k předání informace o tom co člověk viděl. V tomto smyslu lidé v historii využívali malby a obrazy až po vynález fotoaparátu. S nástupem digitalizace a rozvojem technologií zažila oblast práce s obrazem obrovský rozmach. V dnešní době obraz vzniká nejčastěji pomocí techniky jako jsou fotoaparáty, kamery, ale i průmyslová nebo lékařská zařízení.

Práci s obrazem dnes pokrývá mnoho oborů, které se navzájem překrývají a doplňují. Mezi takové můžeme zařadit počítačovou grafiku, strojové vidění a další obory zabývající se zpracováním a analýzou obrazu. Praktické využití těchto oborů zasahuje do nejrozličnějších odvětví lidské činnosti, ovlivňují náš každodenní život a mohou ovlivnit i naše zdraví.

Práce s obrazem se hojně využívá v oblasti biomedicíny, kde existuje množství zařízení, jenž ukládají informace ve formě obrazu. Příkladem takových zařízení může být RTG, CT, mikroskopy, termovize a mnoho dalších. Kvalita a čitelnost takových obrazů může být klíčová pro správné určení diagnózy, což v konečném důsledku může mít přímý vliv na zdraví pacienta. Kvalitu těchto obrazů ovlivňuje velké množství faktorů od nedokonalostí pořizovacího zařízení přes vlivy prostředí a podobně. Z tohoto důvodu je nutné obraz zpracovávat a lékaři poskytnout co nejčitelnější obraz.

Jednou z metod, kterou lze na obraz aplikovat je takzvaná detekce hran. Jedná se o jednu z nejzákladnějších technik pro detekci objektů v obraze, která se využívá při analýze obrazu, hledání vzorů v obraze a počítačovém vidění. Detekce hran slouží ke zvýraznění hran v obraze a tím může oddělit důležité prvky v obraze od pozadí a šumu. Detekce hran zvýrazňuje důležité prvky v obraze. Pro detekci hran existuje mnoho metod, tyto metody nazýváme hranové detektory. V této práci jsou popsány především detektory aproximující první a druhou derivaci. Jsou to metody Sobel, Roberts, Prewitt, Canny, Laplacian a Zerocross. Z nekonvenčních metod je popsána metoda využívající fuzzy logiku. Detektory podávají různé výsledky na obraze bez šumu.

Cílem této práce je implementace experimentů, které nám umožní určit nejvhodnější metodu detekce hran pro obrazy zašumělé různými typy šumů. Dále také porovnat vlastní a nativní implementaci funkce pro detekci hran a zjistit, zda jednotlivé operátory fungují stejně při změně vstupní datové sady. Tato práce popisuje základní úvod do zpracování obrazu a důležité pojmy v této tematice. Dále popisuje základní principy segmentace obrazu, mezi které patří také hlavní téma této práce, detekce hran. Vybrané detektory hran byly nastudovány a teoreticky popsány.

2 Úvod do zpracování obrazu

Obraz můžeme intuitivně chápat jako odraz reálného světa. Takový obraz je zachycen na sítnici oka nebo snímacích čípech zařízení jako jsou fotoaparáty nebo kamery. Digitální zpracování obrazu je disciplína, která se zabývá zpracováváním těchto informací, získaných jako data z fotoaparátu, kamery nebo například různých lékařských zobrazovacích technik.

2.1 Proces zpracování obrazu

Obraz je spojitou optickou veličinou a jeho zpracování je možné rozdělit do několika kroků, kde konkrétní implementace jednotlivých kroků záleží na dané aplikaci.

- Snímání a digitalizace
- Předzpracování
- Segmentace
- Popis obrazu
- Klasifikace

2.1.1 Snímání a digitalizace

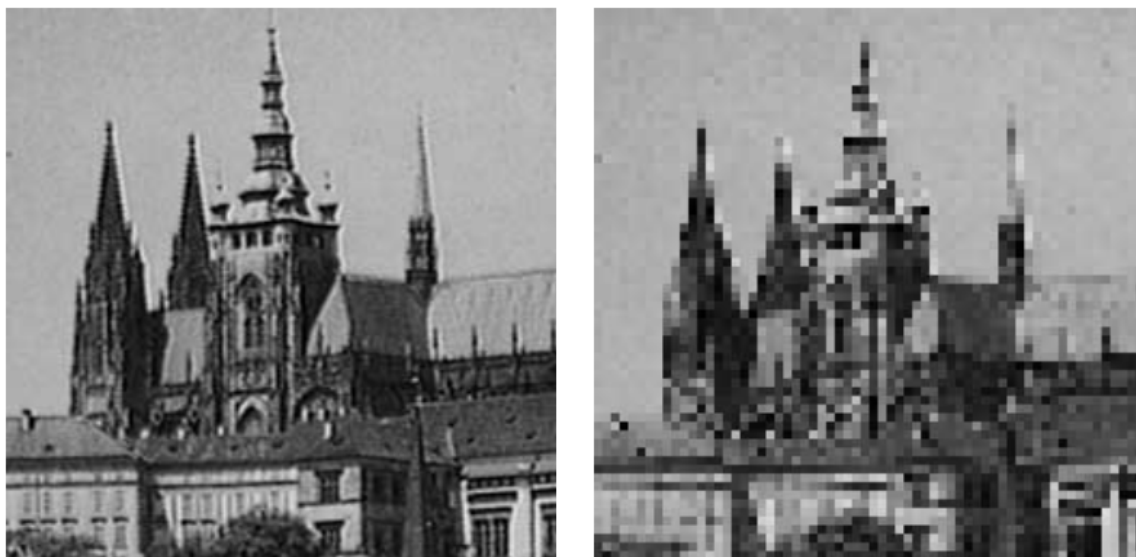
Snímáním obrazu rozumíme převod analogové optické veličiny na elektrický signál. Snímání v elektrických zařízeních obstarává snímací čip, tímto způsobem vznikají obrazy spojitě, což pro digitální zpracování není vhodné. Obraz proto musíme převést do diskrétní podoby, tento proces se nazývá vzorkování [1].

Vzorkování

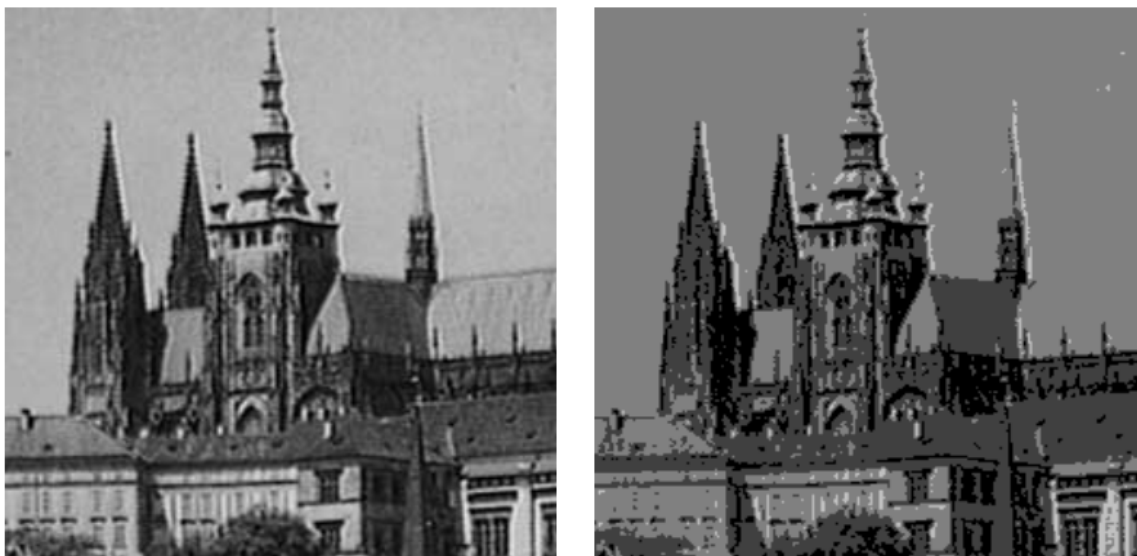
Vstupní signál je popsán funkcí $f(x, y)$ dvou proměnných, které znázorňují souřadnice v obraze, obraz je tedy popsán pomocí matice čísel s N řádky a M sloupci. Hodnota $N \times M$ odpovídá počtu pixelů, na který byl obraz rozložen. Vzorkování se řídí Shannonovým teorémem, který říká, že vzorkovací frekvence musí být minimálně dvakrát větší než je nejvyšší frekvence vzorkovaného signálu, tímto zamezíme tzv. aliasingu, který znemožňuje rekonstrukci obrazu. Naopak při volbě extrémně vysoké vzorkovací frekvence dochází k nárůstu výsledného rozlišení a tím i paměťové náročnosti [2].

Kvantování

Kvantováním nabývá jasová funkce obrazu celočíselných hodnot, tedy úrovní. Počet úrovní musí být dostatečně velký, v opačném případě by mohlo dojít ke zkreslení obrazu a také ke vzniku falešných obrysů. Čím větší je počet pixelů a čím větší je počet jasových úrovní, tím více se digitalizovaný obraz blíží původnímu [2, 3].



Obrázek 1: Vzorkování, vlevo rozlišení 256x256, vpravo 56x56[4]



Obrázek 2: Kvantování, vlevo 256 jasových úrovní, vpravo 4 jasové úrovně[4]

2.1.2 Předzpracování

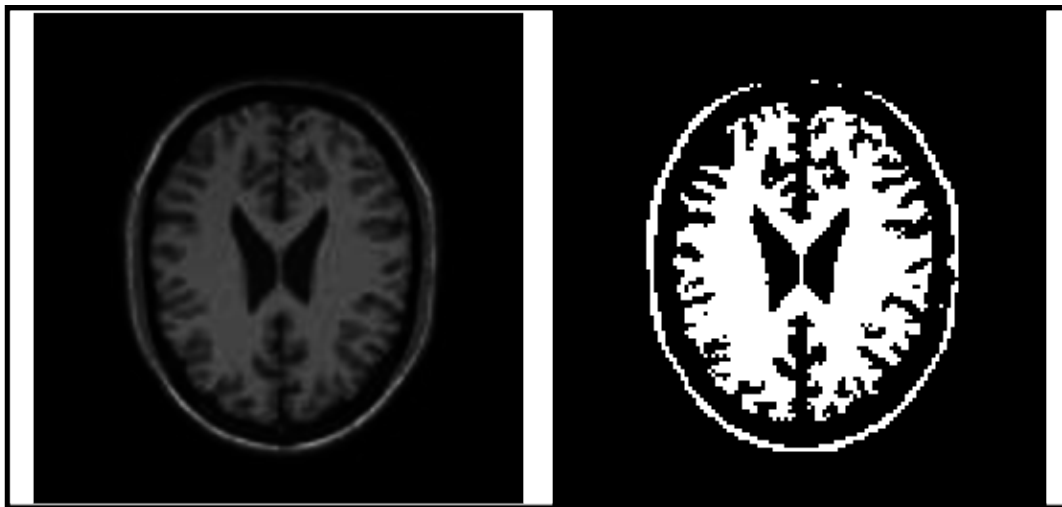
Následuje předzpracování obrazu. Cílem předzpracování je upravení a vylepšení obrazu pro další zpracování. Mezi tyto úpravy patří potlačení šumu, odstranění zkreslení, nebo zvýraznění či potlačení rysů obrazu. Předzpracování obrazu upravujeme na základě toho, co chceme z obrazu získat [3, 4].

2.1.3 Segmentace

Naším cílem není zpracovat obraz jako celek, ale rozpoznat a extrahovat jeho jednotlivé části a objekty. Proces separace objektů obrazu od nezajímavého pozadí se nazývá segmentace obrazu. Existuje mnoho možností jak obraz segmentovat. Segmentace je nejčastěji založena na detekci hran, které ohraničují objekt našeho zájmu nebo také na detekci celých oblastí, kterými jsou objekty v obraze reprezentovány. Pro analýzu obrazu je segmentace jedním z nejdůležitějších kroků. Jedná se tedy o rozdělení obrazu na části. Po segmentaci by měla každá část obrazu představovat jeden celý objekt daného obrazu. V praxi ale dokonalá segmentace není dosažitelná, proto je naší snahou co nejlepší segmentace částečná, kdy objekty obrazu představují oblasti s co nejvyšší homogenitou, tedy oblasti, které mají přibližně stejnou hodnotu jasu [14]. Metody segmentace obrazu :

- Segmentace prahováním

Prahování je nejjednodušší metoda segmentace obrazu, tato metoda hodnotí jasovou hodnotu každého pixelu a hledá takové hodnoty, které mají všechny hodnoty jasu nižší než je práh odpovídající pozadí a všechny hodnoty vyšší než práh popředí. Využití prahování je vhodné u obrazů, kde se výrazně liší objekty obrazu od pozadí [10]. Obrázek číslo 3 znázorňuje segmentaci pomocí prahování s prahem 0.4.

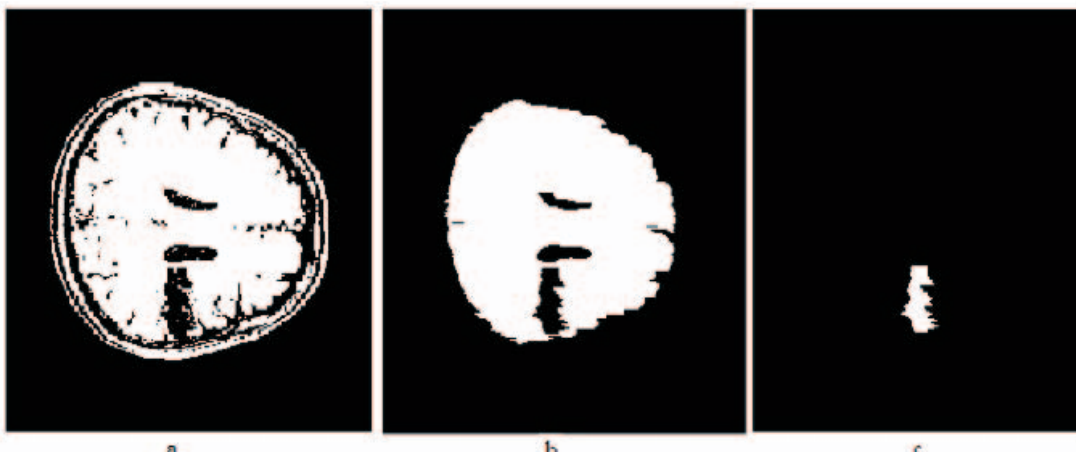


Obrázek 3: Segmentace prahováním[6]

- Segmentace narůstáním oblastí

V případě této metody je obraz rozčleněn do několika homogenních oblastí, z nichž se vybere jeden pixel s určitou jasovou úrovní. Zkoumáme okolní pixely a na základě předem známých atributů homogenity rozhodujeme, zda daný pixel patří do vybrané oblasti nebo nepatří [10].

- Segmentace na základě detekce hran



Obrázek 4: Segmentace obrazu pomocí narůstání oblastí[7]

Segmentace pomocí detekce významných hran v obraze. Toto téma bude podrobně rozebráno v následující kapitole.

2.1.4 Popis obrazu

Segmentací je obraz rozčleněn na jednotlivé oblasti, které jsou následně opatřeny popisem. Popis slouží k jejich kvalifikaci, proto by měl být co nejpřesnější [11]. Popis můžeme rozdělit na :

- kvantitativní: popis pomocí numerických charakteristik, například počet objektů v oblasti nebo velikost oblasti
- kvalitativní: popis tvarů, vztahů mezi oblastmi a jejich vlastnosti

2.1.5 Klasifikace

Posledním krokem v procesu zpracování obrazu je klasifikace. V této fázi jsou popsané oblasti řazeny do tříd pomocí společných prvků. Tento proces můžeme rozdělit do tří tříd. Numerických charekteristik využívá příznaková klasifikace, využívá tedy kvantitativní popis obrazu. S kvalitativním popisem pracuje klasifikace syntaktická, která třídí objekty podle struktury. Třetí třída využívá sloučení předchozích dvou tříd a nazývá se hybridní [4].

2.2 Formát dat

Tato část se zabývá formou, v jaké jsou obrazová data uložena a zpracovávána.

BMP

Jedná se o nekomprimovaný rastový formát, v němž jsou obrazy ukládány po jednotlivých pixelech. Obrazy mají velkou velikost, nejsou tedy vhodné pro zasílání a sdílení. Výhodou tohoto formátu je jednoduchost a volné použití. Pomocí množství bitů, které je použito pro reprezentaci

jenoho pixelu je možné rozlišit různé množství barev, např. 2 barvy - 1 bit na pixel, 256 barev, 8 pixelů nebo 16,7 milionů 24 bitů [9].

JPEG

Joint Photographic Expert Group je formát ztrátově kompresovaných obrazových dat, který je ideální a speciálně navržen pro ukládání fotografií. Míra komprese je u tohoto formátu nastavitelná, proto je vhodný jak pro poměrně náročné aplikace, kde záleží na kvalitě obrazu, tak pro aplikace, kde záleží hlavně na velikosti souboru. Při vyšším kompresním poměru může dojít ke ztrátě informace a ke tvorbě obrazových artefaktů. Výhodou formátu JPEG je jeho rozšířenost, což umožňuje práci s tímto formátem v řadě editorů a aplikací [9].

GIF

Graphics Interchange Format je rastrový formát s bezztrátovou kompresí, tento formát je určen především pro internetovou grafiku, která používá omezenou barevnost. Nehodí se naopak pro ukládání fotografií a to z důvodu omezeného počtu barev [9].

PNG

Portable Network Graphics je také formát s bezztrátovou kompresí, který byl navržen jako náhrada za formát GIF. Na rozdíl od formátu GIF umožňuje PNG plnohodnotné uložení barev a je vhodný jak pro grafiku, tak pro archivaci obrazů. Dále formát umožňuje průhlednost obrazových bodů, jejíž míru označujeme jako alfa kanál. V kombinaci s barevným formátem RGB mluvíme o tzv. RGBA. Průhledný bod nabývá barvu pozadí [9].

DICOM

Digital Imaging and Communications in Medicine je datový standard využívaný pro manipulaci, ukládání, tisk a přenos biomedicínských obrazových dat získaných z rentgenu, magnetické rezonance, atd [13].

2.2.1 Rastrová a vektorová data

U rastrové grafiky je obraz popsán pomocí pixelů, nejmenších obrazových bodů, které jsou uspořádány do pravoúhlé mřížky, jejichž plošná hustota určuje rozlišení obrazu. Paměť počítače obsahuje informaci o velikosti jasu pro každý obrazový bod (pixel). Je-li rozlišení obrazu vysoké jsou jednotlivé pixely pro lidské oko nerozlišitelné a vytváří souvislou plochu, naopak je-li rozlišení malé, jsme schopni vnímat jednotlivé pixely a obraz je kostrbatý. Vektorová grafika rozděluje obraz na matematicky definované části, body, přímky, křivky a mnohoúhelníky. Všechny části vektorového obrazu mají přiřazenu kódovanou informaci o barvě, tloušťce a typu čáry. Nároky na paměť počítače jsou u tohoto typu dat minimální, tudíž jde jejich parametry snadno změnit.

2.2.2 RGB, CMYK a monochromatická data

RGB

Pro vytvoření obrazu jak u rastrové, tak u vektorové grafiky je třeba míchání barev. Jedním ze základních modelů je RGB, který je využíván v zobrazovacích zařízeních. Využívá tří základních barev, red(červená), green(zelená), blue(modrá), jejichž sloučením vytvoří téměř všechny barvy viditelného spektra. Tento model využívá aditivního míchání barev, tedy kombinace barev 0 0 0 vytváří černou barvu a se zvyšující hodnotou se barvy mění a po maximální hodnotě 1 1 1, jejíž výsledkem je bílá barva. Pokud jsou hodnoty jednotlivých barev stejné, jedná se o odstín šedi. V počítačové grafice se rozsah pro každou barevnou složku pohybuje nejčastěji od 0 do 255, což odpovídá barevné hloubce 8 bitů a celkovému počtu možných kombinací barev více než 16 milionů. V počítačové grafice se často používá zápis v hexadecimálním tvaru např. #66FF66 [5].

CMY a CMYK

Modely CMY a CMYK se používají při tisku. Jedná se o substraktivní neboli rozdílový systém míchání barev, v tomto systému se barvy neskládají, ale odečítají od bílé barvy. Základním modelem je CMY (cyan (azurová), magenta (fialová), yellow (žlutá)), z něhož byl odvozen model CMYK black (černá), který přidává černou barvu, která se v tiskárnách často používá [5].

Vztah mezi RGB a CMY:

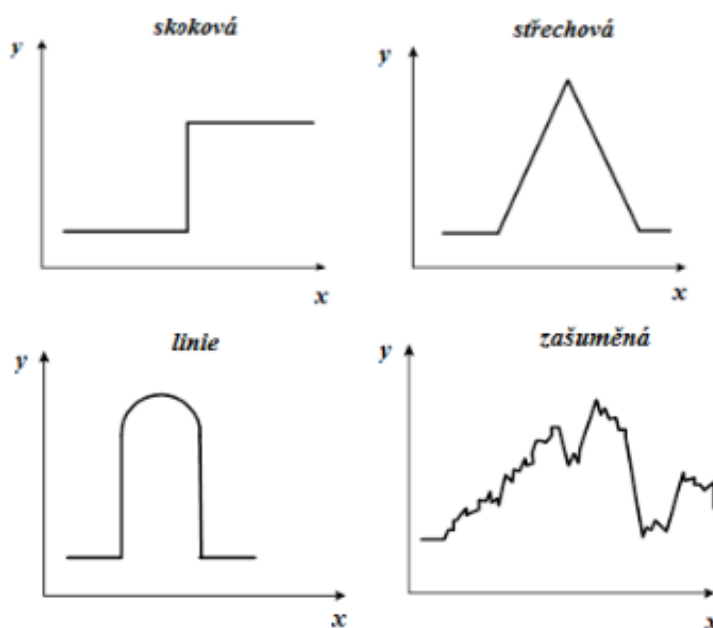
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

Monochromatická data

Monochromatické obrazy jsou obrazy v odstínech jedné barvy nejčastěji v odstínech šedi, tedy v rozmezí bílá - stupně šedi - černá. Ve většině případů pracujeme s 256 odstíny šedi [16, 17].

3 Obecná řešerše matematických metod pro hranové operátory

Hranu můžeme definovat jako místo, kde se výrazně mění jasová funkce sousedících pixelů. Pro výpočet změny funkce se používají parciální derivace. Existují čtyři základní modely hran, skoková, střežová a tenká linie, tyto modely můžeme označit za ideální. V reálném obraze se vyskytují hrany zašumělé, protože v reálných obrazech dochází ke změně jasu obrazu postupně, nikoli skokově. Existuje mnoho hranových detektorů, které využívají různé metody pro detekci hran. V základu můžeme hranové detektory rozdělit na detektory využívající první derivaci nebo druhou derivaci jasové funkce [1, 2].



Obrázek 5: Modely hran[4]

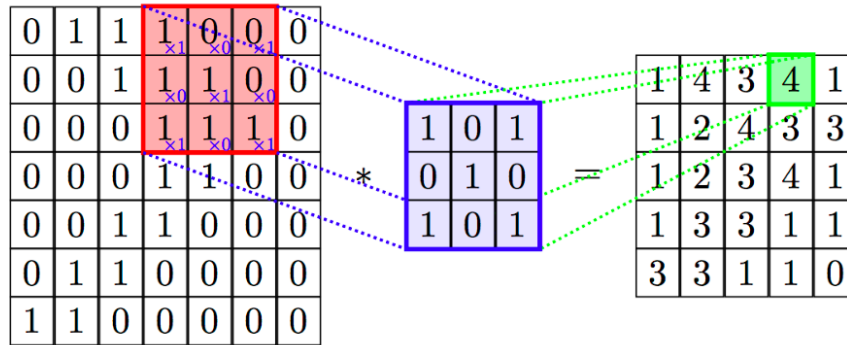
3.1 Konvoluce

Konvoluce je důležitá operace dvou funkcí, a to zpracovávané funkce a jádra. Pro práci s obrazy se využívá tzv. diskrétní konvoluce. Obraz převedený do matice budeme po jednotlivých částech násobit tzv. konvolučním jádrem. Konvoluční jádro je matice o velikosti obvykle od 3x3 do 7x7. Konvoluční jádro budeme postupně posouvat buňku po buňce a hodnoty obrazu pod konvolučním jádrem vynásobíme hodnotami konvolučního jádra, násobky sečteme a uložíme do právě zpracovávané buňky a posuneme se o buňku dále.

Matematický zápis diskrétní konvoluce:

$$I(x, y) = f(x, y) * h(x, y) = \sum_{a=-k}^k \sum_{b=-k}^k f(x-a, y-b) * h(a, b) \quad (2)$$

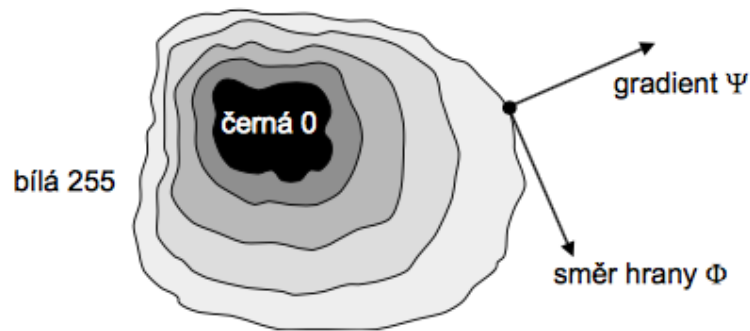
Kde $I(x, y)$ je diskrétní obraz, $f(x, y)$ je intenzita vstupního obrazu a $h(x, y)$ je konvoluční jádro na pozici (a, b) , které také udává její intenzitu [4] [18].



Obrázek 6: Hodnoty jasu obrazu, Konvoluční maska, Výsledný pixel[12]

3.2 Metody detekce hran založené na první derivaci

Absolutní hodnota první derivace má v místě hrany vysokou hodnotu průběhu jasu, tato hodnota popisuje intenzitu kontury v daném bodě. V homogenních oblastech je změna jasu rovna nule, tedy i první derivace je rovna nule. Tyto metody detekce hran nazýváme gradientní metody. Gradient udává změnu funkce, je to vektor daný dvěma složkami, velikostí a směrem. Velikost hrany je velikost gradientu a směr hrany je kolmý ke směru gradientu [1].



Obrázek 7: Směr hrany je kolmý na směr gradientu [4]

Velikost gradientu je dána vztahem :

$$|\nabla g(i, j)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} \quad (3)$$

Směr gradientu je dán vztahem :

$$\psi = \arctan \left(\frac{\frac{\partial g}{\partial y}}{\frac{\partial g}{\partial x}} \right) \quad (4)$$

kde $\arctan(x, y)$ je úhel mezi souřadnou osou x a vektorem k bodu (x, y) . Odpovídají-li oblasti objektu oblastem s homogenním jasnem, jsou body hranice objektu pixely s vysokou hodnotou gradientu, tyto pixely se spojují do hranic. Pro výpočet gradientu používáme hranové operátory, které se dají vyjádřit jako masky pro diskrétní konvoluci.[19][1]

3.2.1 Robertsův hranový detektor

Je jeden z nejstaších hranových operátorů, byl navržen Lawrenceem Robertsem v roce 1965. Zároveň je to jeden z nejjednodušších operátorů. Využívá matici o rozměru 2 x 2 sousedících pixel a stanovuje difference ve dvou na sebe kolmých diagonálních směrech. Hlavní nevýhodou je vysoká citlivost na šum, jelikož využívá malý počet pixelů pro odhad gradientu [4].

Velikost hrany v bodě (x, y) se počítá podle následujícího vzorce:

$$e(x, y) = \sqrt{(f(x, y) - f(x + 1, y + 1))^2 + (f(x + 1, y) - f(x, y + 1))^2} \quad (5)$$

Konvoluční maska má tvar:

$$k_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad k_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

3.2.2 Prewittové hranový detektor

Gradient je odhadován pro osm směrů a využívá konvoluční masku o velikost 3 x 3 pixelů, ale může mít i větší rozměry, např. 5 x 5. Je vybrána vždy maska s největším modulem gradientu. Počítá derivace ve směrech os x, y průměrem, což přispívá k reduci šumu. K výpočtu velikosti hrany v bodě (x, y) využívá hodnot obrazové funkce ve všech okolních pixelech, které pro stručnost označíme [2] :

$$k_1 = \begin{bmatrix} A & B & C \\ D & f(x, y) & F \\ G & H & I \end{bmatrix}$$

Velikost hrany poté vypočítáme pomocí tohoto vzorce :

$$f_x(x, y) = 1/3((C - A) + (F - D) + (I - G)) \quad (6)$$

$$f_y(x, y) = 1/3((A - G) + (B - H) + (C - I)) \quad (7)$$

kde $f_x(x, y)$ je výsledná hodnota velikosti hrany ve směru x a $f_y(x, y)$ je výsledná hodnota velikosti hrany ve směru y .

Konvoluční masky mají tvar:

$$k_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad k_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad k_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

3.2.3 Sobelův hranový operátor

Tento operátor je podobný operátoru Prewittové, liší se tím, že pro výpočet derivací používá vážený průměr. Často využívá pro detekci vodorovných a svislých hran [2].

Velikost hrany v bodě (x, y) se počítá podle následujícího vzorce:

$$f_x(x, y) = 1/4((C - A) + 2(F - D) + (I - G)) \quad (8)$$

$$f_y(x, y) = 1/4((A - G) + 2(B - H) + (C - I)) \quad (9)$$

kde $f_x(x, y)$ je výsledná hodnota velikosti hrany ve směru x a $f_y(x, y)$ je výsledná hodnota velikosti hrany ve směru y .

Konvoluční masky mají tvar:

$$k_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad k_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad k_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

3.2.4 Kirschův operátor

Stejně jako všechny výše uvedené oprátory používá k výpočtu gradientu aproximaci první derivace. Maximální hodnota derivace odpovídá velikosti hrany. Pro stručnost si opět zavedeme náhradní parametry obrazové funkce [4]:

$$k_1 = \begin{bmatrix} A_0 & A_1 & A_2 \\ A_7 & f(x, y) & A_3 \\ A_6 & A_5 & A_4 \end{bmatrix}$$

Veliskot hrany v bodě (x, y) se vypočítá dle vzorce:

$$e(x, y) = \max_{i=0}^7 (|5S_i - 3T_i|) \quad (10)$$

kde

$$S_i = A_i + A_{i+1} + A_{i+2} \quad (11)$$

$$T_i = A_{i+3} + A_{i+5} + A_{i+6} + A_{i+7} \quad (12)$$

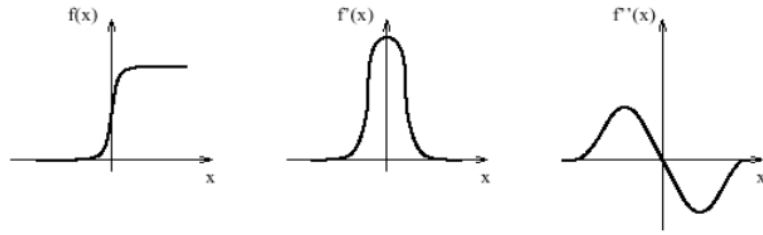
Index i , pro který je ve vztahu (11) dosaženo maxima, určuje směr hrany. Pokud ve vztazích (12) nebo (13) vyjdou hodnoty vyšší než 7 upraví se operací modul o 8.

Konvoluční masky mají tvar:

$$k_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \quad k_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \quad k_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}$$

3.3 Metody detekce hran založené na druhé derivaci

V místě, kde se nachází maximum první derivace protíná druhá derivace nulovou hodnotu, což je základem hledání hrany pomocí druhé derivace. Tato metoda je díky strmosti přechodu mnohem spolehlivější než metody založené na první derivaci. Hlavní nevýhodou těchto metod je větší citlivost na šum. Metody detekce hran založené na druhé derivaci můžeme také nazývat zero-crossing [1].



Obrázek 8: Frűbűh jasové funkce a její první a druhá derivace [15]

Vzorec pro výpočet hrany pomocí druhé derivace:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (13)$$

,kde ∇^2 značí Laplaceův operátor.

3.3.1 Laplaceův operátor

Jedná se o operátor, který aproximuje druhou derivaci. Jedná se o skalár, udává tedy jen velikost hrany, ne její směr, proto je jeho použití vhodné pouze v případě, kdy nám postačí detekovat pouze hranu. Mezi nevýhody patří vysoká citlivost na šum a dvojité odezvy na tenké linie v obraze. Výhodou je, že je vhodný pro ostření obrazu a jeho výpočetní nenáročnost [20].

Laplaceův operátor je definován vztahem :

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (14)$$

V případě, že je obrazová funkce diskrétní, použijeme místo derivací difference :

$$\nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y - 1) - 4f(x, y) \quad (15)$$

Konvoluční maska má tvar :

$$k_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad k_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

V některých případech se používá Laplacián, který využívá větší váhu pixelů blíže reprezentativnímu bodu masky. V takovémto případě ztrácí invariantnost vůči otočení [20].

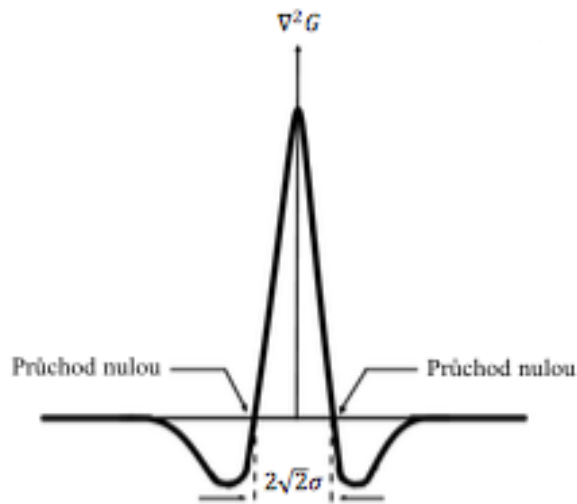
$$k_1 = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix} \quad k_2 = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

3.3.2 Laplacián Gausiánu

Pro přesnější výsledky můžeme před použitím Laplaciánu aplikovat vyhlazující filtr ve tvaru Gaussovy křivky, který redukuje šum. Tento detektor je znám také pod názvem Marr-Hildrethův detektor nebo také pod zkratkou LoG. Nevýhodou tohoto operátoru je přílišné vyhlazování ostrých tvarů, což vede ke ztrátě ostrých rohů [8]. Konvoluční maska má tvar:

$$k_1 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Pro svůj tvar může být maska nazývána mexický klobouk.



Obrázek 9: Příčný řez operátoru LoG
[8]

3.3.3 Cannyho operátor

Byl publikován v roce 1986 Johnem F. Canny. Může být označován jako ideální hranový detektor. Byl navržen tak, aby splňoval tři základní kritéria:

- Minimální chybovost, detektor nemá detekovat falešné hrany, ale bezpečně nalézt všechny podstatné.
- Lokalizace požaduje minimální rozdíl mezi polohou skutečné a detekované hrany.
- Jednoznačná odezva zajišťuje, aby každá hrana byla detekována pouze jednou.

Detekce hran pomocí Cannyho operátoru se realizuje v několika krocích:

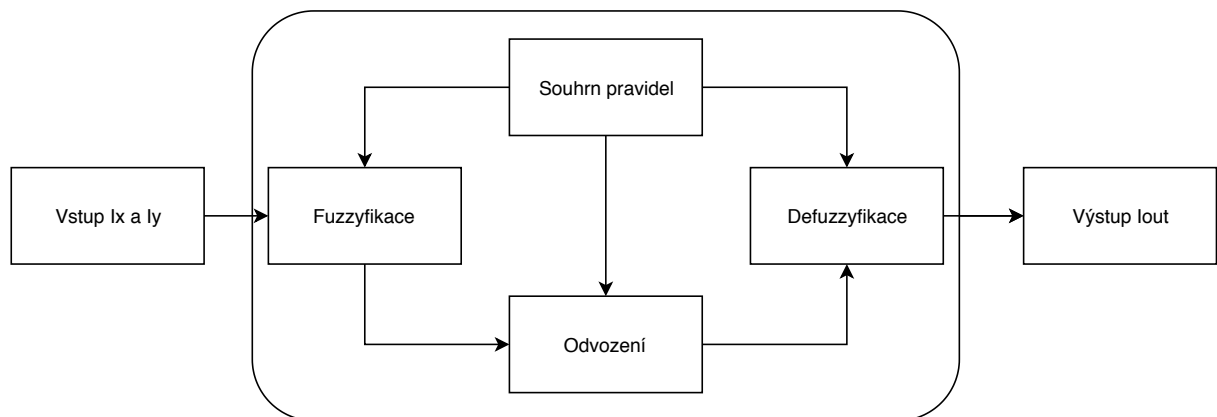
- Eliminace šumu - odstranění šumu ze vstupního obrazu, nejčastěji pomocí Gaussového filtru.
- Určení gradientu - pomocí vhodné konvoluční masky nebo pomocí operátoru první derivace Gaussovske funkce určíme výsledný gradient.
- Nalezení lokálního maxima - v tomto kroku potlačujeme body, které nejsou lokální maxima. Tím získáme jednoznačnou odezvu na hranu a také dojde ke ztenčení hran.
- Eliminace nevýznamných hran - posledním krokem je prahování s hysterezí. Hystereze se používá pro sledování zbývajících pixelů, které nebyly potlačeny. Hystereze používá dvě prahové hodnoty, je-li velikost gradientu pod první prahovou hodnotu, je nastavena na nulu a pokud je nad druhou, vyšší prahovou hodnotu, vytváří hranu automaticky. V případě, že se nachází mezi těmito dvěma prahovými hodnotami, je označen za hranu pouze v případě, sousedí-li s bodem, který již byl jako hrana označen dříve [25].

3.4 Detekce hran využívající fuzzy logiku

Fuzzy logika je rozšířená verze teorie množin ve které neplatí, že prvek pouze patří, nebo nepatří do dané množiny (binární rozhodování), ale zavádí stupeň příslušnosti do určité množiny. Ve fuzzy logice má každý prvek stupeň příslušnosti, jehož hodnota se nachází mezi 1 a 0.

Obecně platí, že techniky fuzzy logiky používané pro detekci hran v obraze jsou založeny na konceptu dělení úrovně šedi obrazu na tři hodnoty. Tyto hodnoty slouží k vytvoření funkce příslušnosti fuzzy systému. Fuzzy systém pracuje jako prahový systém pro zvýšení rozlišování mezi sousedními pixely na hranách obrazu. Metody fuzzy rozpoznávání jsou navrženy pro detekci hran bez stanovení prahové hodnoty. Fuzzy metoda využívá definovaná pravidla, která jsou vytvořena ve formě podmíněného jazykového výrazu logické implikace: jestliže $A = X$, potom $B = Y$, kde X a Y jsou výroky, přičemž X je fuzzy podmínka a Y fuzzy důsledek.

Postup algoritmu začíná vypočtením hodnot gradientů pro osy X a Y . Výsledkem jsou dvě matice které slouží jako vstupy pro část aplikující fuzzy logiku. Pro každý pixel vstupuje do fuzzyfikace I_x a I_y jako hodnoty gradientů v příslušných osách. V kroku fuzzyfikace jsou vstupní hodnoty převedeny z hodnot gradientu na příslušnosti do fuzzy množiny. Následně je podle pravidel rozhodnuto zda pixel patří do uniformního prostoru nebo se jedná o hranu. Posledním krokem je defuzzyfikace, která převede míru příslušnosti na hodnotu výsledného pixelu. Metoda



Obrázek 10: Diagram znázorňující implementaci operátoru využívajícího fuzzy logiku

založená na fuzzy logice nemá definovaný práh, má však několik interních parametrů, které mohou ovlivnit citlivost metody.

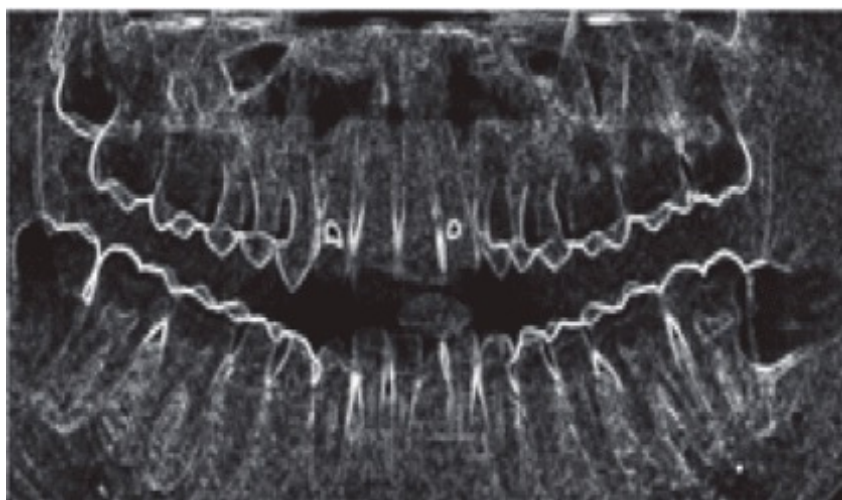
Pro definování fuzzy systému v prostředí MATLAB je potřeba definovat vstupní a výstupní parametry, funkce pro výpočet míry náležitosti v části fuzzyfikace. Takovou funkcí může být například Gaussovská funkce s vhodně definovanou odchylkou a střední hodnotou. Dále je potřeba definovat sadu pravidel a výstupní funkci [24].

4 Biomedicínské aplikace detektorů hran

Segmentace obrazů, jejíž součástí je také detekce hran, hraje klíčovou roli v mnoha lékařských zobrazovacích aplikacích. Automatizuje nebo usnadňuje vymezení anatomických struktur a dalších oblastí zájmu. Detekce hran je využívána v mnoha odvětvích medicíny. Například panoramatický dentální rentgen je široce používán zubními lékaři, protože usnadňuje širší diagnostiku čelisti, zubů a měkkých tkání. Panoramatický snímek vyžaduje zpracování obrazu, které by mohlo vytvářet ostřejší a jasnější obraz nebo by mohlo zlepšit kvalitu informací obsažených v obraze. Segmentace zubů z panoramatických rentgenových snímků může být použita pro extrakci zubu a může hrát důležitou roli v časně diagnostice a léčbě, která pomáhá zubaři lépe rozhodovat [21].



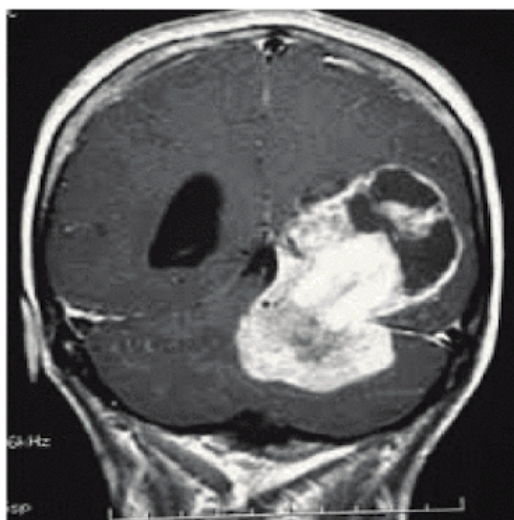
Obrázek 11: Panoramatický rentgenový snímek[21]



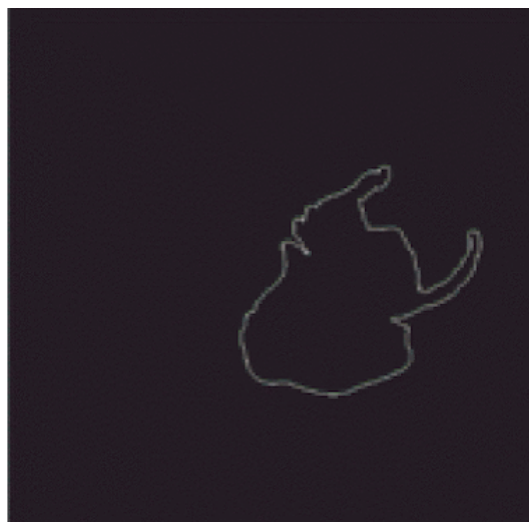
Obrázek 12: Výsledek detekce hran[21]

Dalším využitím detekce hran v biomedicínských obrazech může být detekce nádoru v počá-

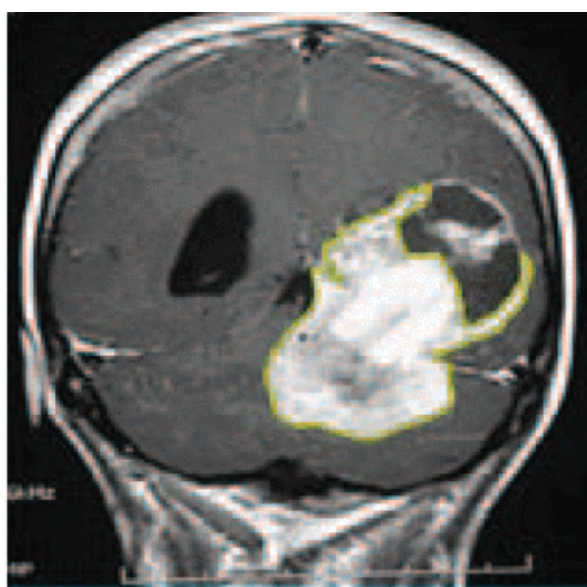
tečným stádiu. Existuje mnoho typů různých nádorů, které se mohou objevit v lidském těle. Nádor je detekován pomocí zdravotnických prostředků, pomocí technik, jako je pozitronová emisní tomografie (PET), počítačová tomografie (CT), zobrazování magnetickou rezonancí (MRI) [22].



Obrázek 13: Původní snímek z MRI[22]



Obrázek 14: Hrany tumoru [22]



Obrázek 15: Původní obraz s vyznačenými hranami tumoru[22]

5 Cíle modelování detekce hran

Pro analýzu efektivity jednotlivých metod je potřeba provést několik kroků, které nám umožní porovnat jednotlivé metody. Mezi testované metody patří : Sobel, Roberts, Prewitt, Canny, Laplacian a Zerocross. Z nekonvenčních metod byla zvolena implementace metody Sobel s využitím fuzzy logiky. Tyto metody budou vzájemně porovnávány na různých datových sadách s různými typy šumů a následně budou jejich výsledky porovnány. Toto testování slouží k porovnání těchto metod a jejich odolnosti vůči šumu. Budou porovnávány výsledky střední kvadratické chyby a korelace detekce hran na zašumělém a nezašumělém obraze. Při porovnání všech metod bychom měli být schopni určit nejlepší metodu a zjistit, jak jsou jednotlivé metody odolné vůči dynamické změně šumu.

6 Testování detektorů hran na reálných medicínských datech

V této části budou popsány testované detektory implementované funkcí *edge* v prostředí MATLAB. Všechny dostupné detektory této funkce byly testovány na obrazech zašumělých čtyřmi různými šumy, které budou popsány v následující kapitole. Vstupem pro tuto funkci je obraz v monochromatickém formátu a výstupem je tento obraz s hranami detekovanými pomocí zvolené metody. Metoda detekce hran je jedním z parametrů této funkce, defaultním operátorem je metoda Sobel. V této práci byly zkoumány pouze dva parametry této funkce a to metoda pro detekci hran a práh. Funkce *edge* má v MATLABu následující syntaxi :

$$I_{out} = edge(I),$$

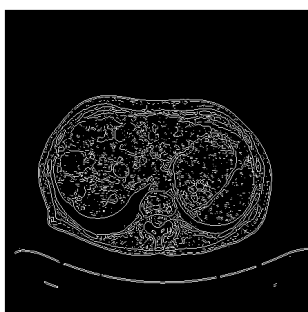
kde I je vstupní obraz a I_{out} je výstupní obraz s detekovanými hranami.

6.1 Metoda Sobel

Při použití funkce *edge* bez stanovení jakýchkoli parametrů je výsledkem detekce hran pomocí metody Sobel s práhem 0.1772 pro horizontální i vertikální hrany. Následující obrázek zobrazuje detekci hran na nezašumělém obraze a obraze zašumělém šumem Salt & Pepper s použitím Sobelova operátoru s různým nastavením práhu. Syntaxe kódu v MATLABu je následující:

$$I_{out} = edge(I, 'sobel', thresh);,$$

kde I je vstupní obraz a I_{out} je výstupní obraz s detekovanými hranami, *sobel* je zvolená metoda pro detekci hran a *thresh* je zvolená hodnota práhu. Prah může být od 0 do 1, avšak při práhu 0.3 a vyšším na testovaných obrazech již hrany nebyly viditelné.



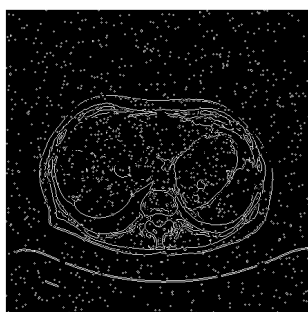
(a)



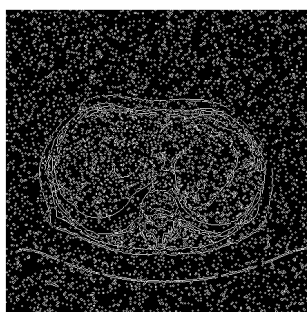
(b)



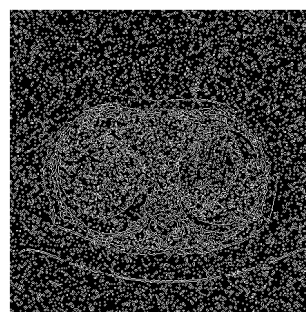
(c)



(d)



(e)



(f)

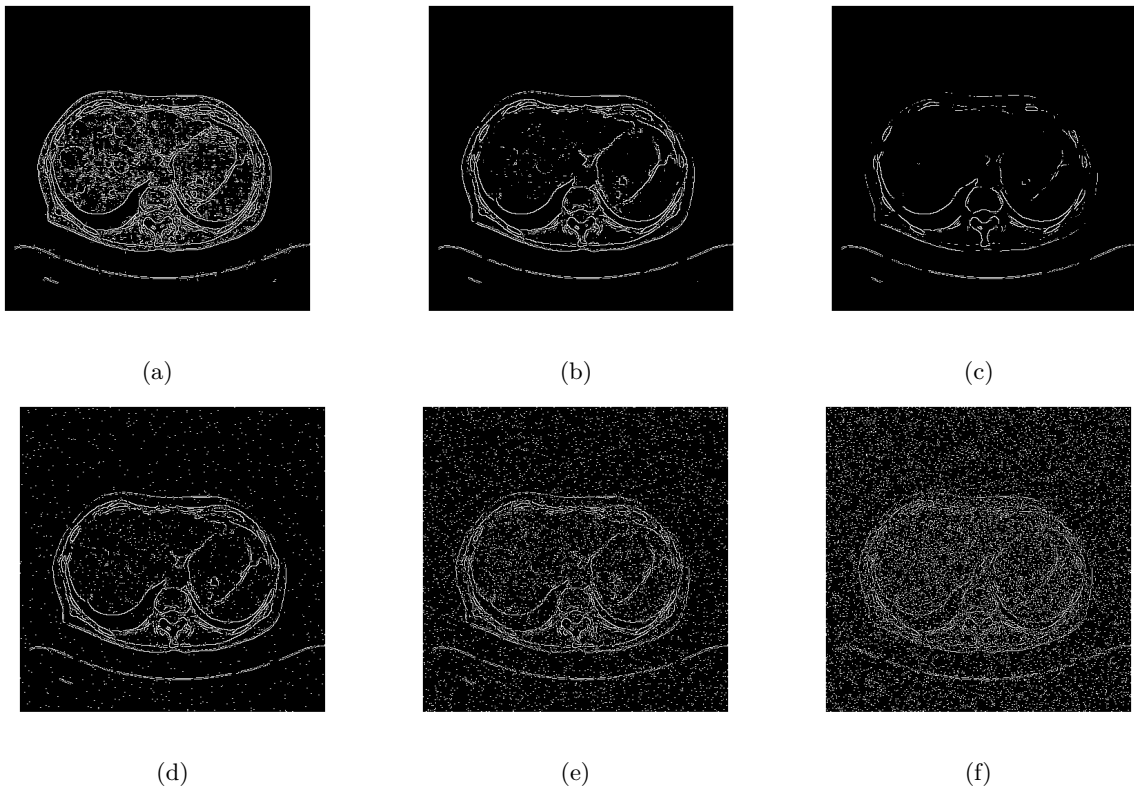
Obrázek 16: Detekce hran s použitím Sobelova operátoru na nezašumělém obraze s práhem a) 0.025, b) 0.05, c) 0.1 a na obraze zašumělém šumem Salt & Pepper s hustotou šumu d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašumělém obraze byl použit práh 0.05

6.2 Metoda Roberts

Dalším hranovým detektorem funkce *edge* je metoda Roberts, jehož výpočetní náročnost je ze všech nejmenší a to z důvodů velikosti jeho konvoluční masky 2×2 pixelů. Syntaxe v MATLABU:

$$I_{out} = edge(I, 'roberts', thresh);$$

kde I je vstupní obraz a I_{out} je výstupní obraz s detekovanými hranami, *roberts* je zvolená metoda pro detekci hran a *thresh* je zvolená hodnota práhu. Následující obrázek zobrazuje detekci hran pomocí Robertsova operátoru pro tři různé práhy na nezašumělém snímku, další tři obrázky zobrazují detekci hran s práhem 0.05 na zašumělém snímku se třemi různými parametry šumu.



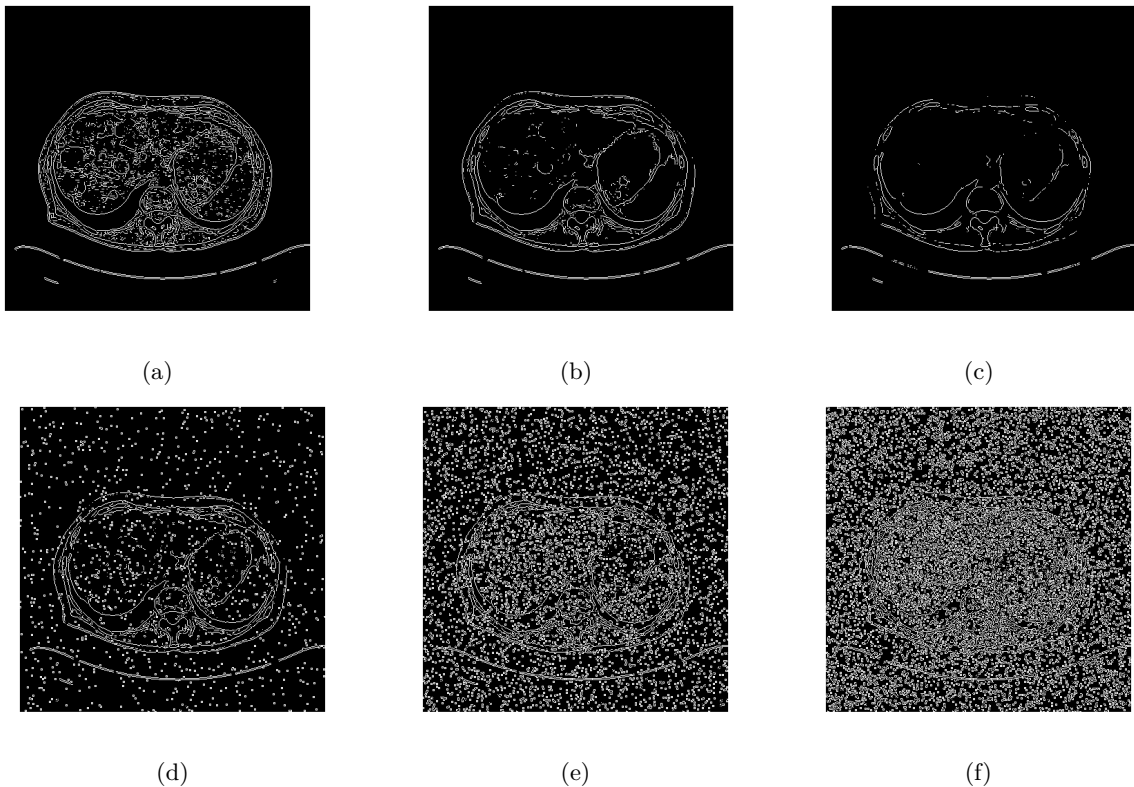
Obrázek 17: Detekce hran s použitím operátoru Roberts na nezašumělém obraze s práhem a) 0.025, b) 0.05, c) 0.1 a na obraze zašumělém šumem Salt & Pepper s hustotou šumu d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašumělém obraze byl použit práh 0.05

6.3 Metoda Prewitt

Nastavení detektoru Prewitt se nijak výrazně neliší od metody Sobel. Hlavním rozdílem je použitá konvoluční maska. Syntaxe v MATLABU:

$$I_{out} = \text{edge}(I, 'prewitt', \text{thresh});$$

kde I je vstupní obraz a I_{out} je výstupní obraz s detekovanými hranami, *prewitt* je zvolená metoda pro detekci hran a *thresh* je zvolená hodnota práhu. Následující obrázek zobrazuje detekci hran pomocí operátoru Prewitt pro tři různé práhy na nezašumělém snímku, další tři obrázky zobrazují detekci hran s práhem 0.05 na zašumělém snímku se třemi různými parametry šumu.



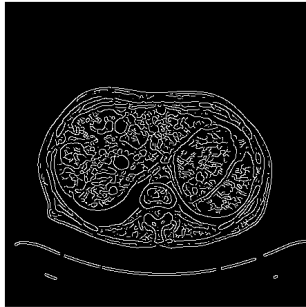
Obrázek 18: Detekce hran s použitím operátoru Prewitt na nezašumělém obraze s práhem a) 0.025, b) 0.05, c) 0.1 a na obraze zašumělém šumem Salt & Pepper s hustotou šumu d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašumělém obraze byl použit práh 0.05

6.4 Metoda Canny

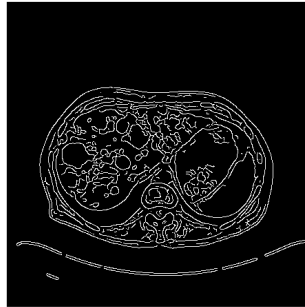
Cannyho detektor se od ostatních liší prahováním, využívá prahování s hysterezí, viz kapitola 3.3.3. Detekuje jemné hrany pouze v případě, že jsou spojeny s výraznějšími hranami, což zabraňuje detekování nežádoucího šumu jako hrany. V MATLABu jej zapíšeme :

$$I_{out} = \text{edge}(I, 'canny', \text{thresh});,$$

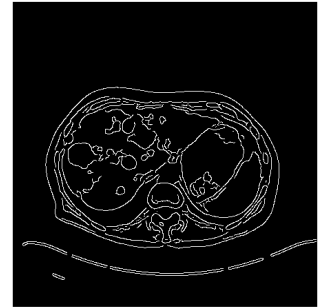
kde I je vstupní obraz a I_{out} je výstupní obraz s detekovanými hranami, *canny* je zvolená metoda pro detekci hran a *thresh* je zvolená hodnota práhu.



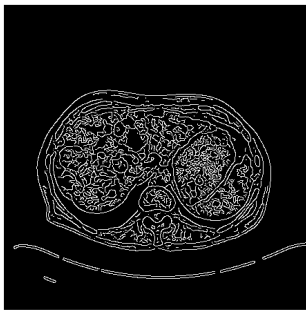
(a)



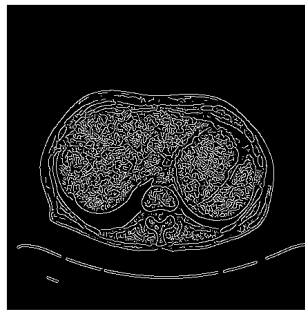
(b)



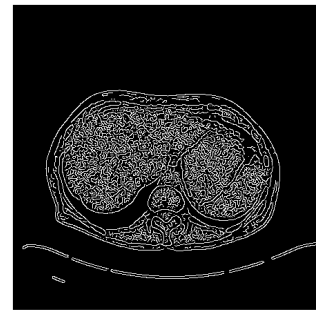
(c)



(d)



(e)



(f)

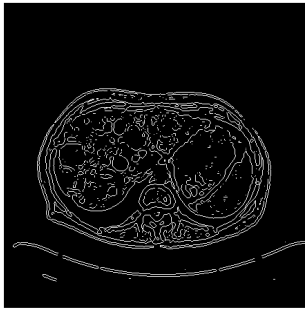
Obrázek 19: Detekce hran s použitím operátoru Canny na nezašumělém obraze s práhem a) 0.025, b) 0.05, c) 0.1 a na obraze zašumělém šumem Speckle s rozptylem šumu d) 0.01, e) 0.05, f) 0.1 pro detekci hran na zašumělém obraze byl použit práh 0.05

6.5 Metoda Laplacián

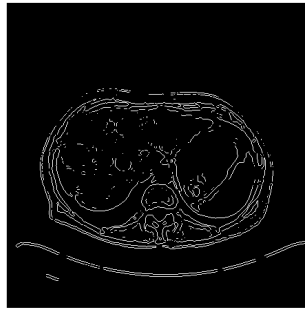
Metoda Laplacián se od předchozích metod liší tím, že aproximuje druhou derivaci, hledáme místa kde druhá derivace prochází nulou. V MATLABu jej zapíšeme:

$$I_{out} = edge(I, 'log', thresh);,$$

kde I je vstupní obraz a I_{out} je výstupní obraz s detekovanými hranami, log je zvolená metoda pro detekci hran a $thresh$ je zvolená hodnota práhu.



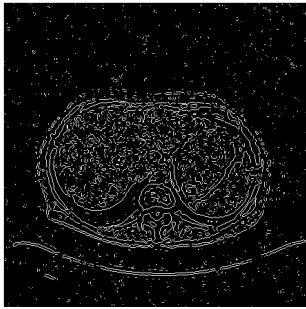
(a)



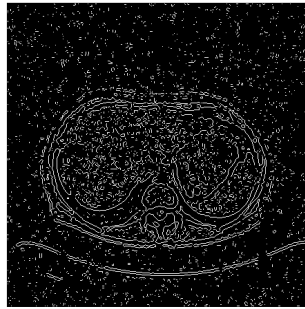
(b)



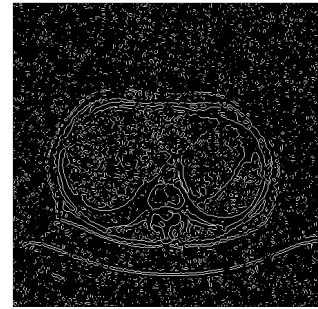
(c)



(d)



(e)



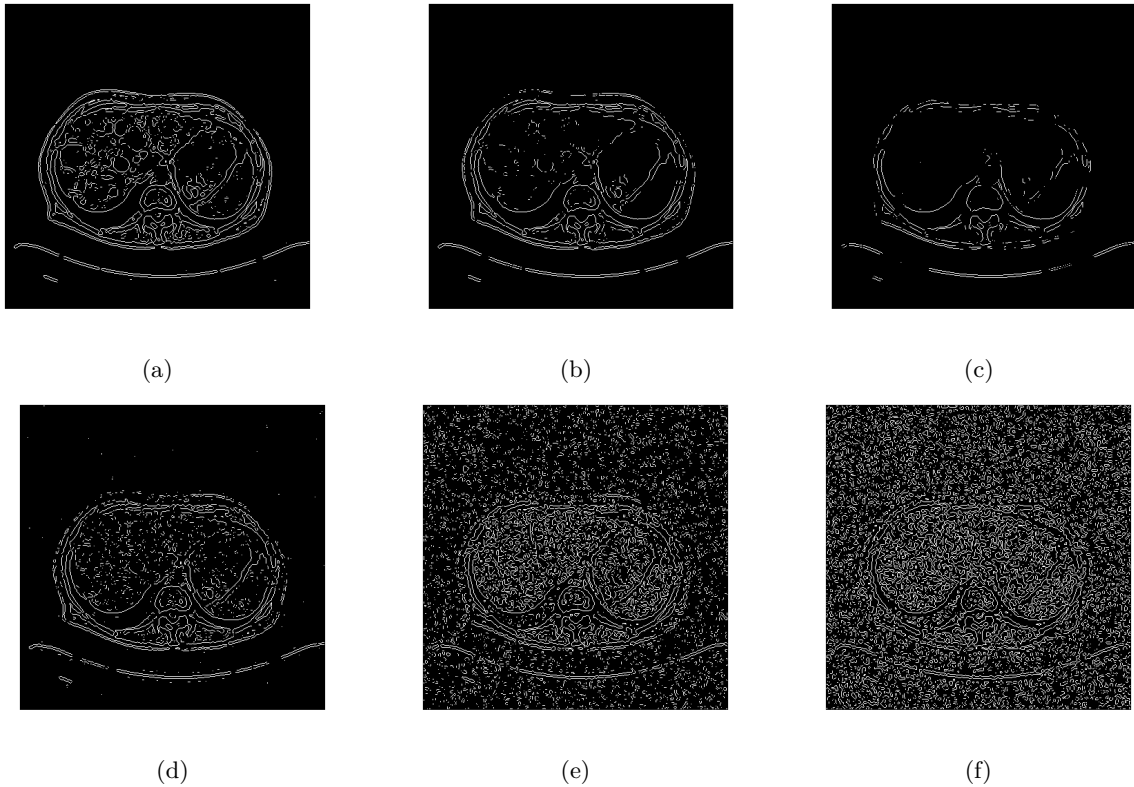
(f)

Obrázek 20: Detekce hran s použitím operátoru Laplacián na nezašumělém obraze s práhem a) 0.0025, b) 0.005, c) 0.01 a na obraze zašumělém Gaussovským šumem s rozptylem d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašumělém obraze byl použit práh 0.005

6.6 Metoda Zero-cross

Zero-cross je metoda založená stejně jako předchozí metoda na průchodu druhé derivace jasové funkce nulou. U této metody specifikujeme *zero-crossing* filtr, v této práci byl použit defaultní Laplaceův filtr. Proto jsou výsledky této metody totožné s výsledky metody Laplacián. Syntaxe v MATLABu je následující: $I_{out} = \text{edge}(I, 'zerocross', \text{tresh});$,

kde I je vstupní obraz a I_{out} je výstupní obraz s detekovanými hranami, *zerocross* je zvolená metoda pro detekci hran a *tresh* je zvolená hodnota práhu.



Obrázek 21: Detekce hran s použitím operátoru Zero-cross na nezašumělém obraze s práhem a) 0.0025, b) 0.005, c) 0.01 a na obraze zašumělém šumem Localvar s rozptylem d) 0.01, e) 0.05 , f) 0.1 pro detekci hran na zašumělém obraze byl použit práh 0.005

7 Šumové generátory

Obrazový šum je náhodné rozmístění jasové, či barevné informace a obvykle je aspektem elektronického šumu. Může být produkován senzorem a obvody digitálního fotoaparátu nebo skeneru. Obrazový šum je nežádoucí vedlejší produkt zachycení obrazu, který přidává nepravdivé a cizí informace. Velikost obrazového šumu se může pohybovat v rozmezí od nepostřehnutelných teček, až po zcela zašumělé snímky, ze kterých téměř nejsme schopi získat jakékoliv informace [23].

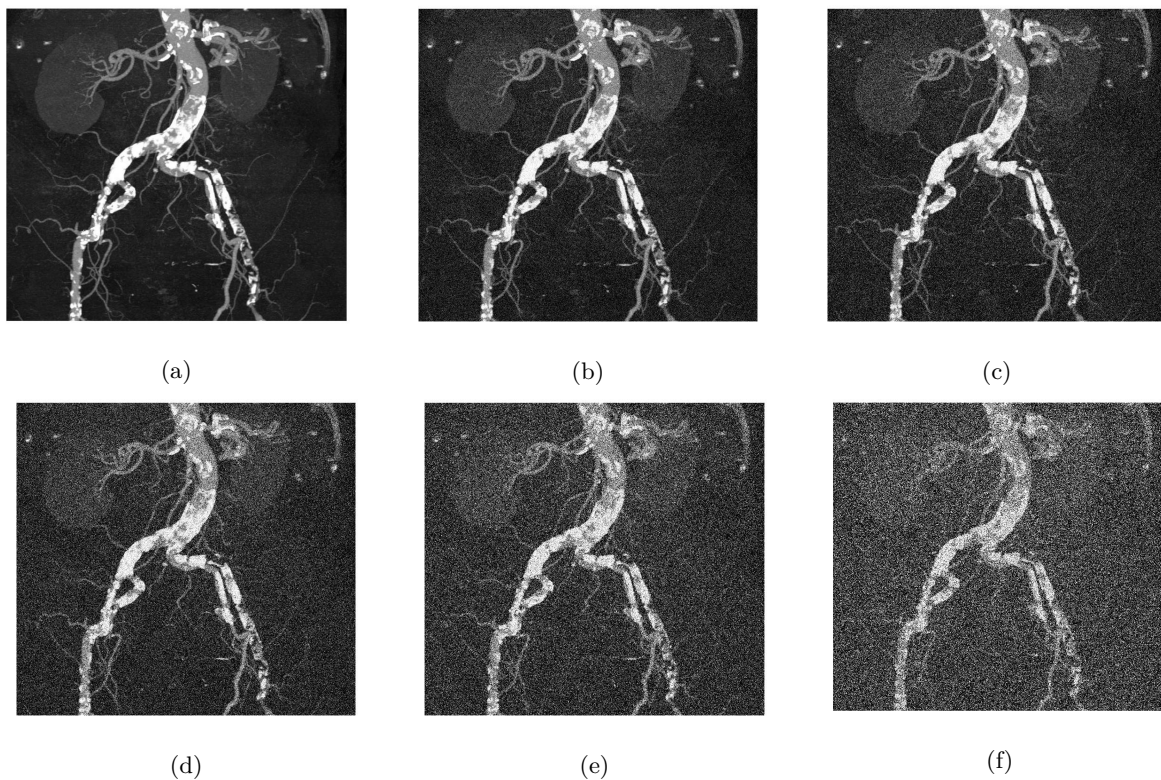
7.1 Gaussovský šum

Gaussovský šum v digitálním obraze vzniká nejčastěji při pořizování snímku. Je důsledkem např. špatného osvětlení nebo vysoké teploty senzoru. Standartní model tohoto šumu je aditivní, nezávislý na každém pixelu a intenzitě signálu. Gaussovský šum může být redukován použitím prostorového filtru [23].

Základní funkcí pro přidání šumu je v MATLABu funkce *imnoise*, která má pro Gaussovský šum následující syntaxi:

$$J = \text{imnoise}(I, 'Gaussian', m, \text{var_gauss}),$$

kde I je název původního obrazu ve stupních šedi, m je průměr gaussovského šumu a var_gauss je rozptyl šumu. J je název výsledného zašumělého obrazu, který má stejnou velikost jako původní obraz I .



Obrázek 22: Gaussovský šum s parametrem rozptylu s hodnotou a) původní obraz ,b) 0.01, c) 0.025, d) 0.05, e) 0.1, f) 0.25

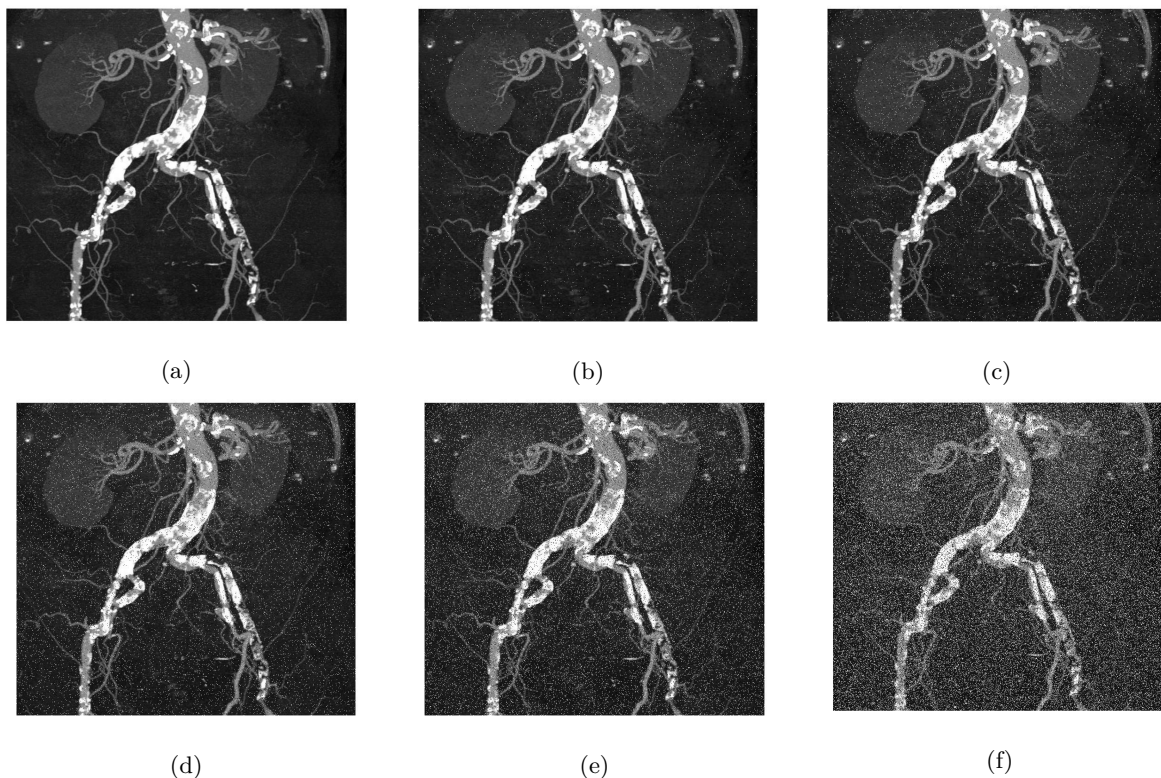
7.2 Salt & Pepper

Šum Salt & Pepper, také znám jako impulsní šum, se prezentuje jako řídce se vyskytující bílé a černé pixely. Tento typ šumu může být způsoben chybami analogově-digitálního převodníku, bitovými chybami v přenosu atd. [23]

Pro přidání tohoto šumu do obrazu v MATLABu použijeme tuto syntaxi :

$$J = \text{imnoise}(I, 'salt\&pepper', d),$$

kde I je název původního obrazu ve stupních šedi, d je hustota šumu a J je název výsledného zašumělého obrazu, který má stejnou velikost jako původní obraz I .



Obrázek 23: Šum Salt & Pepper s parametrem hustoty šumu s hodnotou a) původní obraz ,b) 0.01, c) 0.025, d) 0.05, e) 0.1, f) 0.25

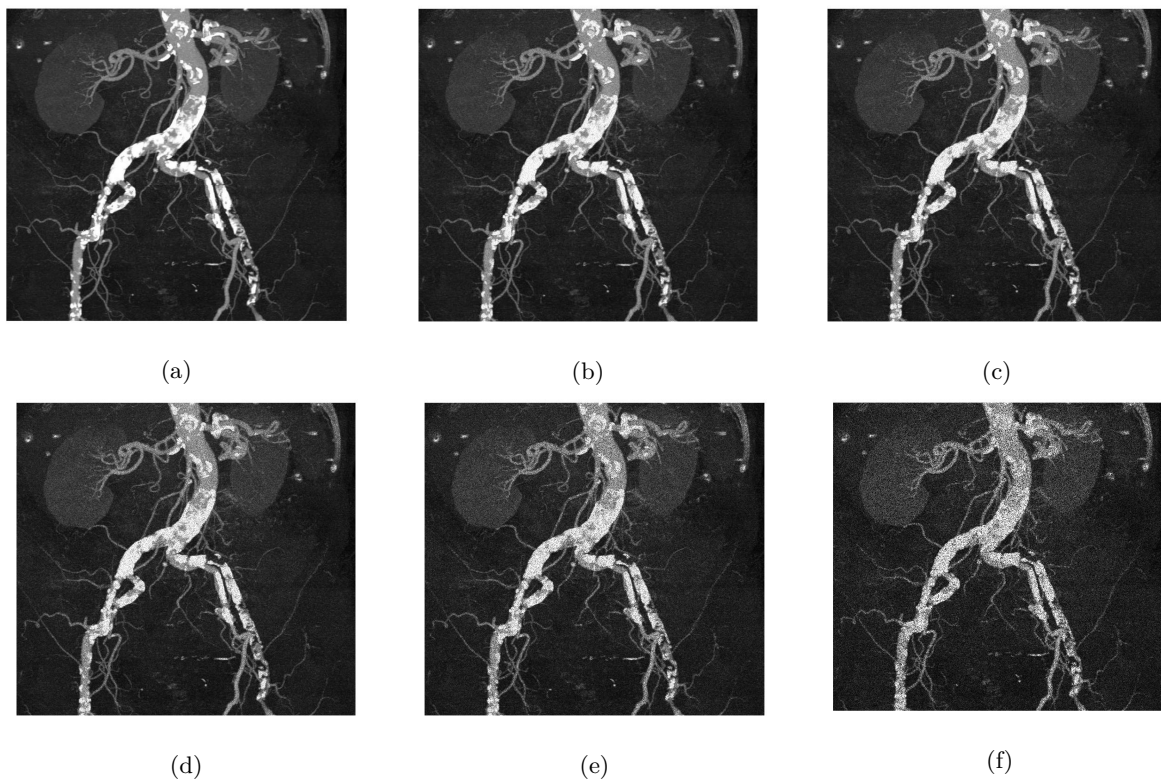
7.3 Speckle

Speckle je zrnitý šum, který degraduje jemné detaily a vymezení hran a omezuje rozlišení kontrastu tím, že ztěžuje detekci malých a nízkých kontrastních lézí v těle. Tímto šumem jsou nejčastěji poškozeny obrazy získané koherentními zobrazovacími systémy, jako je laser nebo ultrazvuk. Modeluje se násobením náhodného čísla hodnotou jasové funkce [1].

V MATLABu jej do obrazu přidáme takto :

$$J = imnoise(I, 'speckle', var_speckle),$$

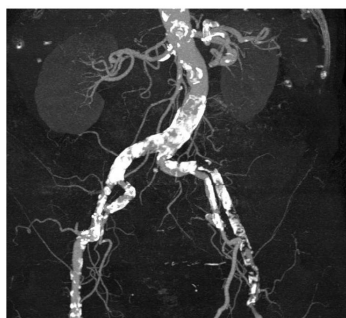
kde I je název původního obrazu ve stupních šedi, $var_speckle$ je rozptyl multiplikativního šumu. J je název výsledného zašumělého obrazu, který má stejnou velikost jako původní obraz I .



Obrázek 24: Šum Speckle s parametrem rozptylu s hodnotou a) původní obraz ,b) 0.01, c) 0.025, d) 0.05, e) 0.1, f) 0.25

7.4 Localvar

Šum Localvar přidává do obrazu Gaussovský šum s nulovou střední hodnotou.



(a)



(b)



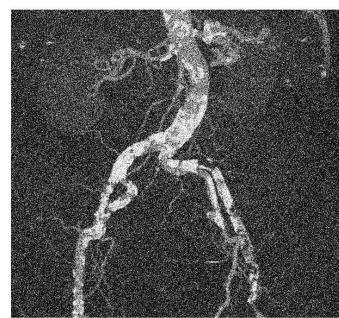
(c)



(d)



(e)



(f)

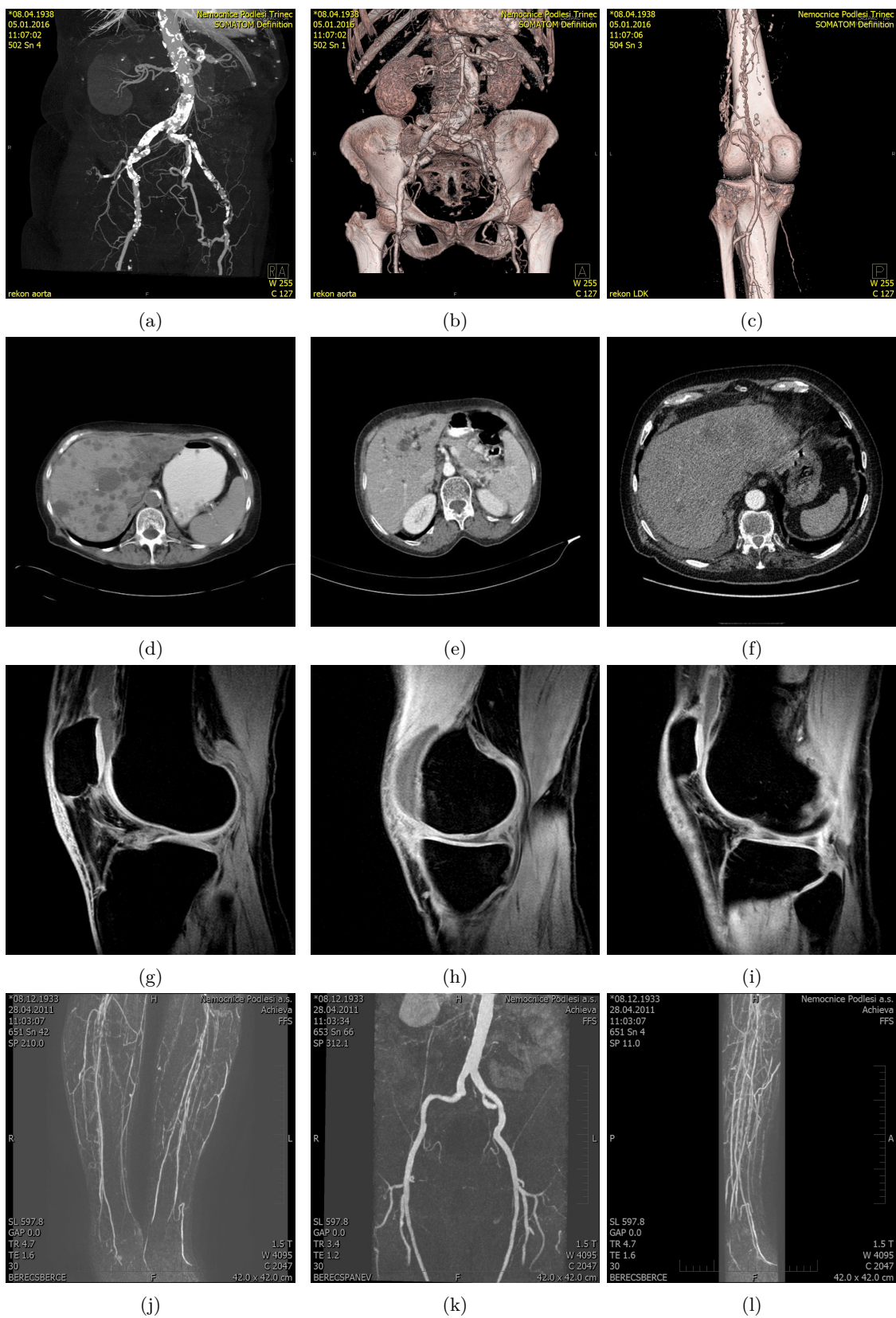
Obrázek 25: Gaussovský šum s parametrem rozptylu s hodnotou a) původní obraz ,b) 0.01, c) 0.025, d) 0.05, e) 0.1, f) 0.25

8 Design experimentu

Tyto experimenty sledují dynamiku efektivity detekce hran v závislosti na intenzitě šumu, kvalitu detekce hran při změně datové sady porovnávají nativní funkci MATLABu pro detekci hran s vlastní implementací této funkce. Výsledky experimentu byly vyhodnocovány pomocí střední kvadratické chyby a korelace.

8.1 Popis datových sad

Snímky použity pro tuto práci byly poskytnuty vedoucím práce. Jedná se o čtyři datové sady obsahující 2812 snímků o celkové velikosti 368MB. První datová sada obsahuje pět pacientů a celkem 126 snímků. Jedná se o snímky kalcifikace krevních cév vzniklé na výpočetní tomografii, pocházející z nemocnice Podlesí Třinec z roku 2015-2016. Všechny obrazy v tomto datasetu mají rozměry 1024 x 1024 a jsou formátu jpg. Další datová sada obsahující pouhých 31 snímků neznámého původu zobrazuje játra, taktéž pomocí výpočetní tomografie. Snímky mají různé rozměry, nejčastěji však 630 x 630 a 442 x 442 a jsou ve formátu jpg. Datová sada zobrazující kloubní chrupavky, pořízené pomocí magnetické rezonance obsahuje 18 snímků ve formátu png o rozměrech 512 x 512. Snímky jsou neznámého původu, pořízeny v roce 2014. Největší datová sada obsahuje 25 pacientů a celkem 2636 snímků zobrazující krevní cévy pomocí magnetické rezonance. Snímky vznikly v nemocnici Podlesí Třinec v letech 2010-2012 a jsou různých rozměrů formátu jpg. Většina snímků z poslední datové sady je v odstínech šedi, ostatní datové sady obsahují obrazy ve formátu RGB, proto byl pro práci s nimi nutný předvod do monochromatického formátu.



Obrázek 26: Ukázka datových sad a)-c) CT - kalcifikace cév , d)-f) CT- játra, g)-i) MRI - kloubní chrpavka, j)-l) MRI - krevní cévy

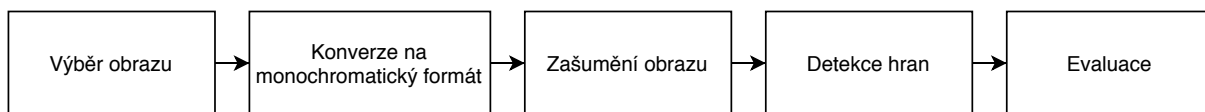
8.2 Konverze obrazu

Obrazy použitých datových sad jsou barevného modelu RGB, který však není pro detekci hran vhodný, proto je nutný převod obrazů do monochromatického formátu. Pro tento převod byla použita funkce MATLABu *rgb2gray*, která odstraňuje informaci o odstínu a sytosti barvy při zachování jasů. Tato funkce převádí hodnoty RGB na hodnoty ve stupních šedi vytvořením váženého součtu komponent R,G,B:

$$x = 0.299 * R + 0.587 * G + 0.114 * B \quad (16)$$

8.3 Průběh jednoho experimentu

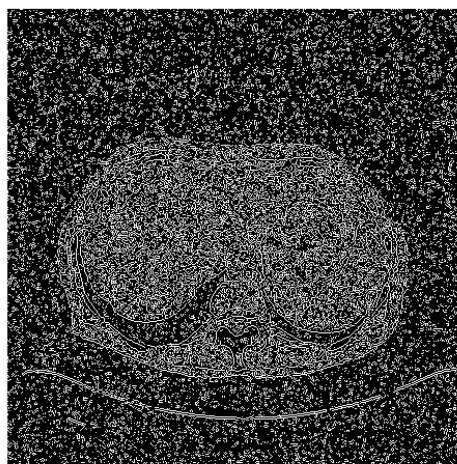
Pro jeden experiment je zvolen jeden z dostupných snímků ,který je nutné pro detekci hran převést do monochromatického formátu. Poté je tento snímek zašuměn pomocí jednoho šumového generátoru se zvolenými parametry a na něm je provedena detekce hran pomocí zvoleného operátoru se zvoleným prahem. Výsledek detekce hran na tomto zašuměném snímku je poté porovnán s detekcí hran na tomtéž snímku bez zašumění a výsledkem je jejich vzájemný vztah vyjádřen pomocí střední kvadratické chyby a korelace. Obrázek číslo 27 zobrazuje ukázkou jednoho experimentu.



Obrázek 27: Vývojový diagram jednoho experimentu



(a)

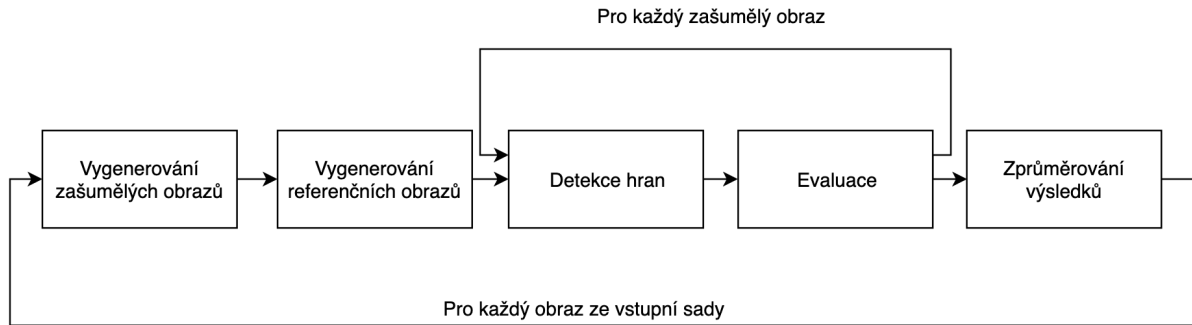


(b)

Obrázek 28: Detekce hran na a) nezašuměném a b) zašuměném obraze s Gaussovským šumem

8.4 Aplikace pro vykonání experimentů

Cílem této aplikace je zjednodušení a urychlení práce s obrazem. Tato aplikace umožňuje spuštění detekce hran pro několik obrazů, jejichž výsledky poté zprůměruje. Také umožňuje zašumění obrazu různými šумы s různými parametry najednou v několika cyklech a provede operaci detekce hran s danou metodou pro různé práhy v jednom běhu aplikace.



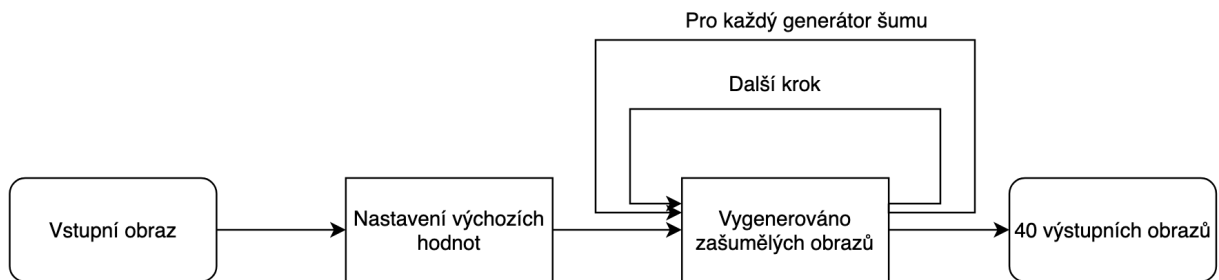
Obrázek 29: Vývojový diagram aplikace pro provedení experimentů

Před spuštěním aplikace je nutné nastavit tyto parametry:

- Vybraný hranový detektor
- Požadovaný počet zašumělých obrazů pro jeden typ šumu

Po spuštění je zvolen jeden z datasetů. Z celého datasetu je následně možné vybrat buďto konkrétní obrazy, nebo celou složku obrazů, které budou zpracovávány. Celý průběh aplikace je znázorněn na obrázku 29

Další část aplikace probíhá v cyklu pro každý obraz ze vstupní sady. Obraz je převeden do monochromatického formátu a vstupuje do části aplikace, která obstarává dynamické generování šumu znázorněné na diagramu 30.



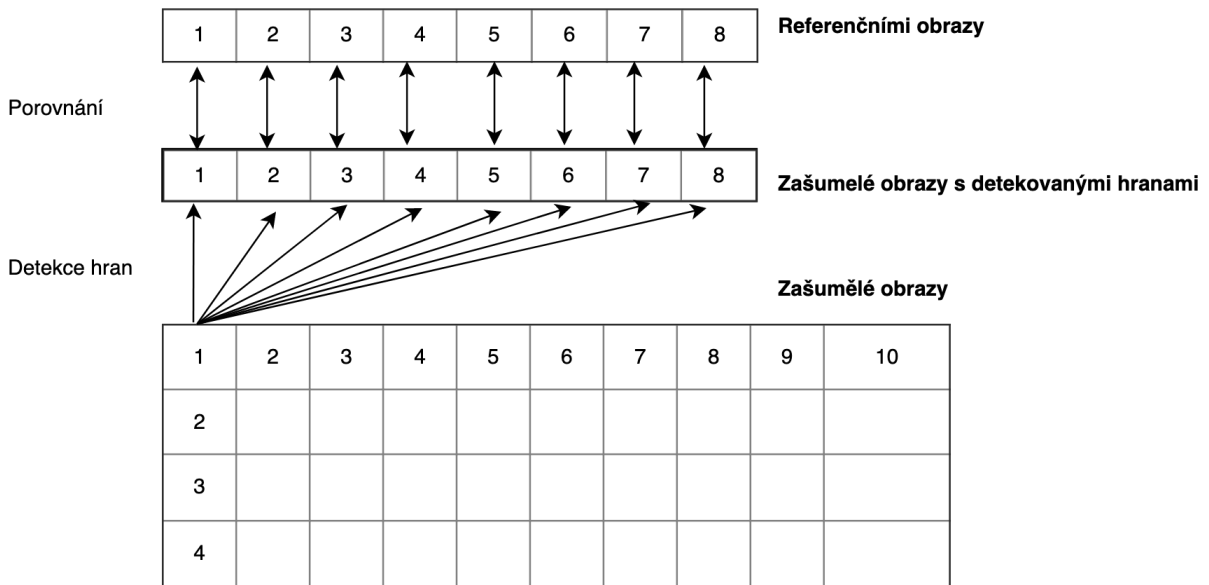
Obrázek 30: Vývojový diagram aplikace pro provedení experimentů

Při nastavení výchozích hodnot je maximální hodnota parametru šumu nastavena jako $max = 0.25$, při použití vyšší hodnoty parametru šumu se výsledné hodnoty detekce hran již nijak výrazně nemění, obraz je pro lidské oko téměř nečitelný. Na základě vstupního parametru

$X = 10$ se provede cyklus o 10 krocích, kde v každém je vygenerován zašumělý obraz s parametrem šumu $p = (0, 25/X) * i$, kde X je počet kroků a i je aktuální krok. Stejný cyklus se provede pro každý ze 4 generátorů šumu. Jelikož zašumění provádíme pomocí 4 generátorů po dokončení generování je výstupem celkem 40 zašumělých obrazů, 10 pro každý generátor šumu.

Poté jsou vytvořeny referenční obrazy, což jsou nezašumělé obrazy s detekovanými hranami pomocí zvolené metody pro detekci hran. Tyto obrazy jsou vytvořeny s prahy od 0.025 do 0.2 s krokem 0.025. Pro metody Laplacian a ZeroCross jsou použity prahy od 0.0025 do 0.02 s krokem 0.0025.

Následujícím krokem je detekce hran zašumělých obrazů a jejich následné srovnání s referenčními obrazy. Tento proces je znázorněn na obrázku 31. Aplikace provede pro každý zašumělý obraz detekci hran s 8 úrovněmi prahů, tím vznikne 8 zašumělých obrazů s detekovanými hranami. Takto vytvořená skupina obrazů vstupuje spolu s referenčními obrazy do funkce pro evaluaci.



Obrázek 31: Vývojový diagram aplikace pro provedení experimentů

V evaluační funkci jsou srovnány obrazy zašumělé s detekovanými hranami s obrazy referenčními. Ve srovnání se srovnávají obrazy, které vznikly s nastavením stejného prahu. Výstupem jsou dvě pole, která obsahují korelační koeficient a střední kvadratickou chybu. Aplikace následně průměruje hodnoty korelačního koeficientu a střední kvadratické chyby podle typu šumu, parametru šumu a použitého prahu pro každý obraz.

Výsledkem celého běhu aplikace jsou dvě 3D matice, jedna s výsledky střední kvadratické chyby a druhá s výsledky korelace. Řádky zaznamenávají jednotlivé šумы, sloupce parametr šumu a listy zaznamenávají jednotlivé prahy. Výsledkem jsou tedy dvě 3D matice o rozměrech

4 x 10 x 8. Tabulka zaznamenává střední kvadratickou chybu pro metodu Sobel s práhem 0.025 :

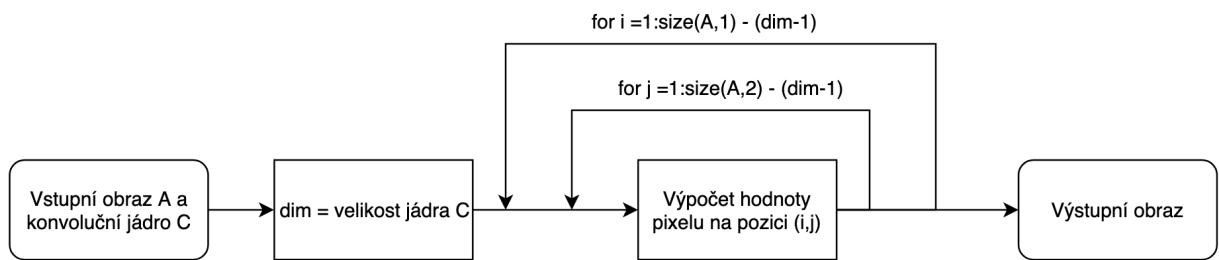
Tabulka 1: Ukázka výsledků experimetu

Šum	Parametr šumu								
	0.025	0.05	0.075	0.1	0.125	0.15	0.175	0.2	0.225
Gaussovský	0.3740	0.3822	0.3850	0.3867	0.3879	0.3889	0.3897	0.3904	0.3910
Salt&Pepper	0.0651	0.1185	0.1628	0.1997	0.2311	0.2569	0.2788	0.2974	0.3132
Speckle	0.0821	0.1056	0.1183	0.1263	0.1319	0.1362	0.1393	0.1420	0.1440
Localvar	0.3475	0.3688	0.3751	0.3782	0.3802	0.3815	0.3827	0.3835	0.3844

Aplikace byla spuštěna pro všechny dostupné operátory funkce edge a také pro vlastní implementaci oprátorů Sobel, Roberts a Prewitt na všech dostupných datových sadách. Výsledky byly uloženy pro pozdější porovnání.

8.5 Vlastní implementace hranových oprátorů

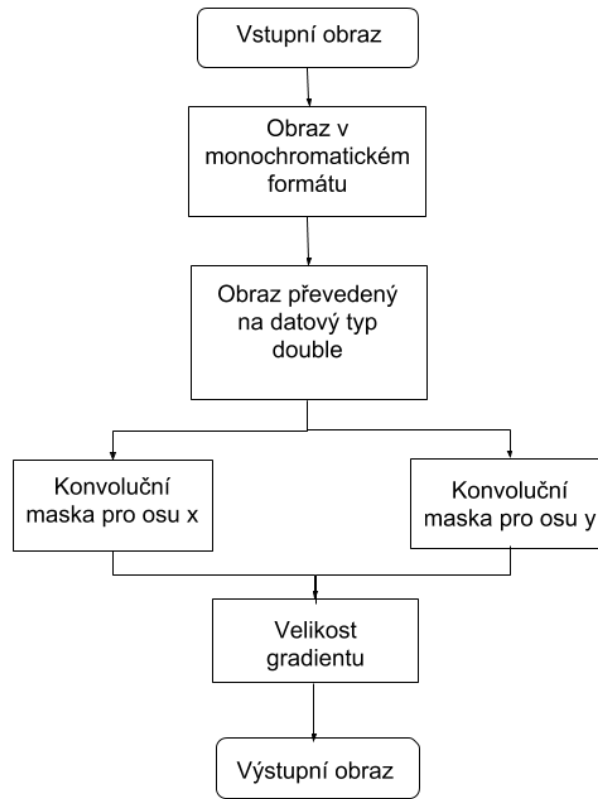
Pro porovnání účinnosti nativní funkce pro detekci hran MATLABu byla vytvořena vlastní implementace hranových operátorů Sobel, Roberts a Prewitt. Implementace byla provedena s použitím nativní funkce MATLABu pro konvoluci, funkce *conv2*, a také s vlastní implementací této funkce. Vlastní implementace konvoluce je znázorněna na diagramu 32. Vstupem pro konvoluci je obraz *A* ve formě matice a konvoluční jádro označené jako *core*. Následně je zjištěna velikost konvolučního jádra a aplikace vstupuje do dvou cyklů, které procházejí matici obrazu. Uvnitř cyklu je proveden výpočet hodnoty na souřadnicích *i* a *j*.



Obrázek 32: Vývojový diagram pro vlastní konvoluci

Pro zjednodušení a zvýšení přehledosti byl vytvořen vývojový diagram 33, který znázorňuje implementaci operátoru pro detekci hran s použitím nativní funkce pro konvoluci.

Postup implementovaného algoritmu je následující. Vybraný obraz, který budeme zpracovávat načteme do pracovní plochy MATLABu. Pro následnou práci je nutný převod do monochromatického formátu, který se provede pomocí funkce *rgb2gray*. Dalším krokem je převod obrazové matice na datový typ double. Následně je podle zvoleného operátoru zvolena konvoluční maska pro osy *x* a *y*. V případě použití nativní funkce MATLABu pro konvoluci jsou hodnoty G_x a



Obrázek 33: Vývojový diagram pro implementaci hranového operátoru

G_y vypočítány pomocí funkce *conv2*, je nutná definice konvolučních masek. V případě vlastní implementace konvoluce se velikost první derivace pro osu x a y pro metodu Sobel vypočítá podle následujícího vzorce:

$$G_x = ((I(x+2, y) + 2 * I(x+1, y+1) + I(x+2, y+2)) - (I(x, y) + 2 * I(x+1, y) + I(x+2, y))) \quad (17)$$

$$G_y = ((I(x, y+2) + 2 * I(x+1, y+2) + I(x+2, y+2)) - (I(x, y) + 2 * I(x+1, y) + I(x+2, y))) \quad (18)$$

,kde I je vstupní obraz. Poté je vypočtena velikost gradientu pomocí vzorce :

$$|\nabla f(x, y)| = \sqrt{(G_x)^2 + (G_y)^2} \quad (19)$$

Výsledkem je binární matice, kde 1 odpovídá hranovým bodům a 0 černému pozadí. Před zobrazením výsledného obrazu je nastaven práh, který je porovnáván s velikostí gradientu. Pokud

je hodnota gradientu vyšší než nastavený práh, bude této hodnotě přiřazena 1 a bude tedy zobrazena na výsledném obraze jako hranový bod, naopak pokud je hodnota gradientu nižší než nastavený práh, bude tento obrazový bod součástí pozadí.

8.6 Evaluační techniky

Pro porovnání podobnosti mezi výsledným zašumělým obrazem s detekovanými hranami a obrazem s hranami bez šumu byly použity dvě evaluační metody, a to střední kvadratická chyba a korelace.

8.6.1 Střední kvadratická chyba

Střední kvadratická chyba je střední hodnota mocnin rozdílů mezi dvěma veličinami, v tomto případě se jedná o rozdíl hodnoty souhlasných pixelů zašumělého obrazu s detekovanými hranami a hranami na nezašumělém obraze. Aby byl výpočet možný, musí mít oba obrazy stejnou velikost. Vzorec pro výpočet střední kvadratické chyby obrazu je následující:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (20)$$

,kde n je počet pixelů obrazu, y_i je původní obraz s detekovanými hranami a y'_i je zašumělý obraz s detekovanými hranami.

Pro výpočet střední kvadratické chyby v MATLABu byla použita funkce *immse*, která počítá střední kvadratickou chybu mezi poli X a Y, které mohou být matice libovolného rozměru, ale musí mít stejnou velikost a třídu. V případě, že jsou obrazy shodné, je střední kvadratická chyba rovna nule.

8.6.2 Korelace

Korelace je statistická technika, která ukazuje, zda existuje a jak silný je, vztah mezi dvěma proměnnými. Hlavní výsledek korelace se nazývá korelační koeficient a jeho rozsah je od -1 do 1. Čím více se korelační koeficient blíží číslu 1 nebo -1, tím je podobnost dvou proměnných vyšší, naopak blíží-li se koeficient nule znamená to, že neexistuje žádný vztah mezi těmito proměnnými. Pro výpočet korelace mezi zašumělým a nezašumělým obrazem s detekovanými hranami byla použita funkce MATLABu *corr2*, která vrací 2D korelační koeficient. Tato funkce k výpočtu využívá tento vzorec :

$$corr(A, B) = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (21)$$

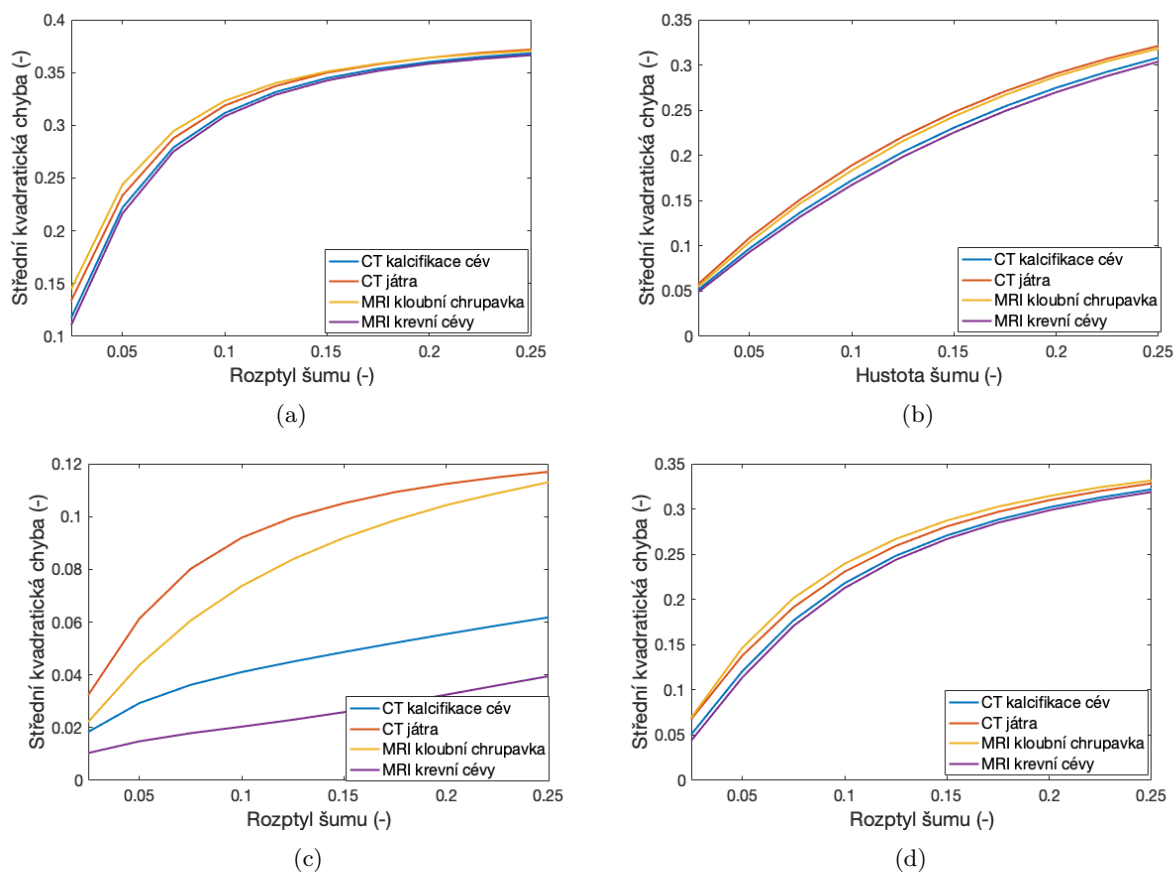
,kde A a B jsou porovnávané obrazy, \bar{A} , \bar{B} jsou střední hodnoty těchto obrazů a m a n jsou čísla řádků a sloupců jednotlivých matic.

9 Výsledky experimentů

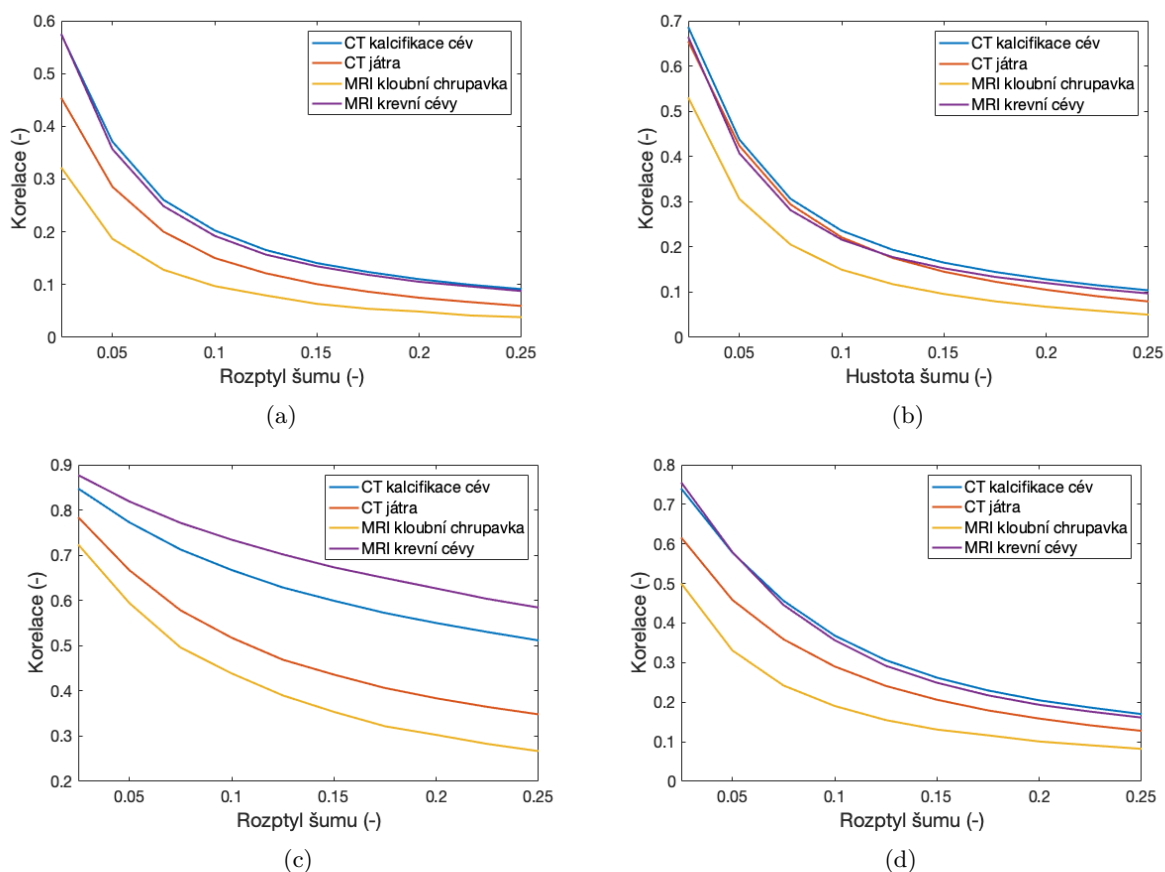
Tyto výsledky vznikly zprůměrováním výsledných hodnot jednotlivých obrazů pro jednu nebo více datových sad, z každé datové sady byla vybrána referenční sada obrazů obsahující 17 - 30 obrazů, podle možností datové sady.

9.1 Porovnání změny chování operátoru při změně datasetu

Tyto experimenty sledují vývoj střední kvadratické chyby při změně datové sady pro metodu Sobel s práhem 0.1, které jsou ovlivněny jednotlivými šumy. Jak můžeme vidět na obrázku 34 u šumů Localvar, Salt & Pepper a Gaussovského šumu se při změně datového setu střední kvadratická chyba nijak výrazně nemění a její trend zůstává téměř stejný. Naopak s použitím šumu Speckle se při změně datového setu střední kvadratická chyba výrazně mění. Mohlo by to být důsledkem toho, že šum Speckle narozdíl od ostatních šumů přidává šum pouze na objekty obrazu, nikoliv do pozadí, proto záleží jak velkou plochu pozadí objekt zakrývá, což se se změnou datasetu mění.



Obrázek 34: Vývoj střední kvadratické chyby při změně datasetu pro metodu Sobel s práhem 0.1 zašumělé šumem a)Gaussovským, b)Salt & Pepper, c)Speckle, d)Localvar



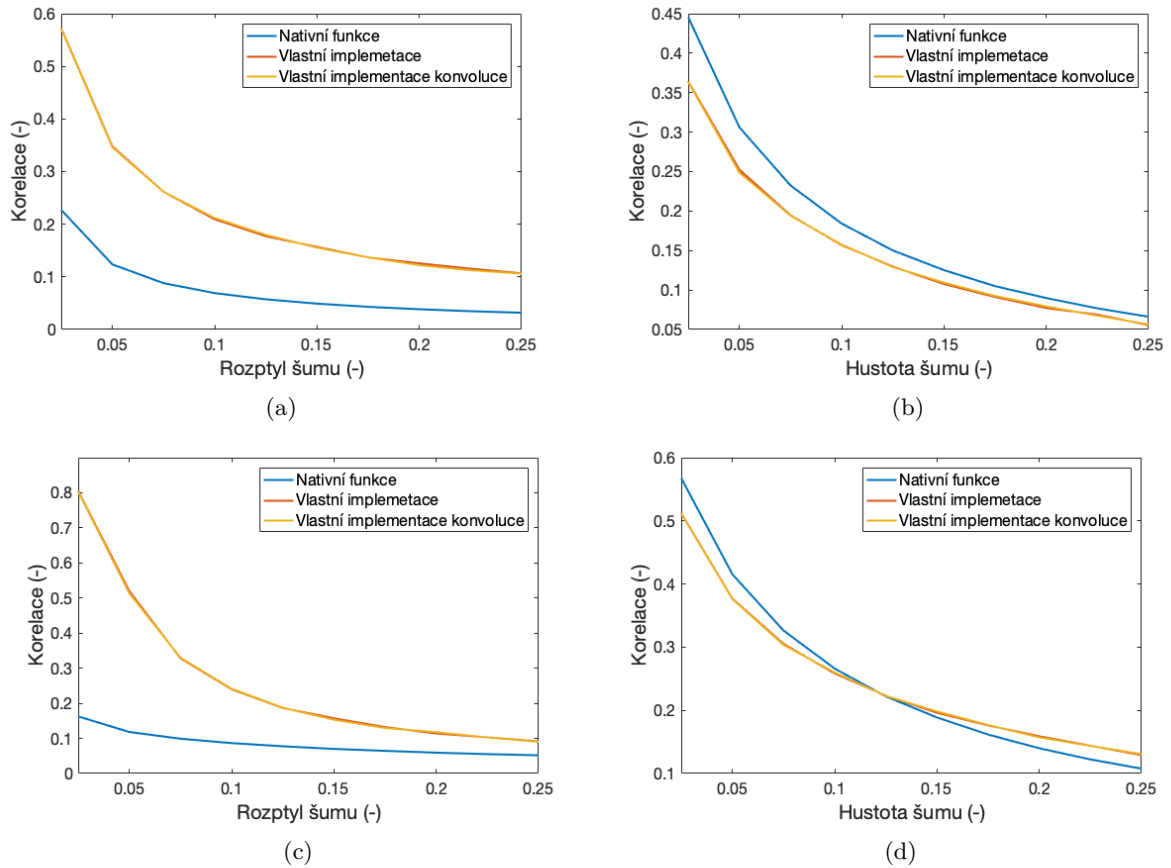
Obrázek 35: Vývoj korelace při změně datasetu pro metodu Laplacián s práhem 0.01 zašumělé šumem a) Gaussovským, b) Salt & Pepper, c) Speckle, d) Localvar

Tyto grafy znázorňují změnu korelace obrazů s hranami detekovanými pomocí metody Laplacián při změně datového setu s různými šumy. Nejmenší změna korelace při změně datového setu nastane při změně datasetu 1 a 4, jedná se o datasety obsahující obrazy krevních cév. Jedná se o velmi podobné datasety, čímž by se dala vysvětlit podobnost vývoje jejich korelačních křivek.

9.2 Srovnání vlastní a nativní implementace

Pomocí těchto experimentů je pozorována změna korelace a střední kvadratické chyby pro metody Sobel, Roberts a Prewitt zašumělé šumem Salt & Pepper nebo Gaussovským šumem s použitím nativní funkce MATLABu pro detekci hran a vlastní implementací této funkce s použitím nativní funkce pro konvoluci, tak i s vlastní implementací této funkce. Grafy ukazují, že detekce hran na obrazech zašumělých Gaussovským šumem je výrazně lepší s použitím vlastní implementace a to jak s použitím nativní funkce pro konvoluci, tak s použitím vlastní implementace této funkce. V případě použití šumu Salt & Pepper se výsledky příliš neliší a v případě použití operátoru Sobel je pro tento typ šumu vhodnější použít nativní funkci pro detekci hran. Při použití vlastní implementace jsou výsledky s použitím nativní funkce pro konvoluci téměř

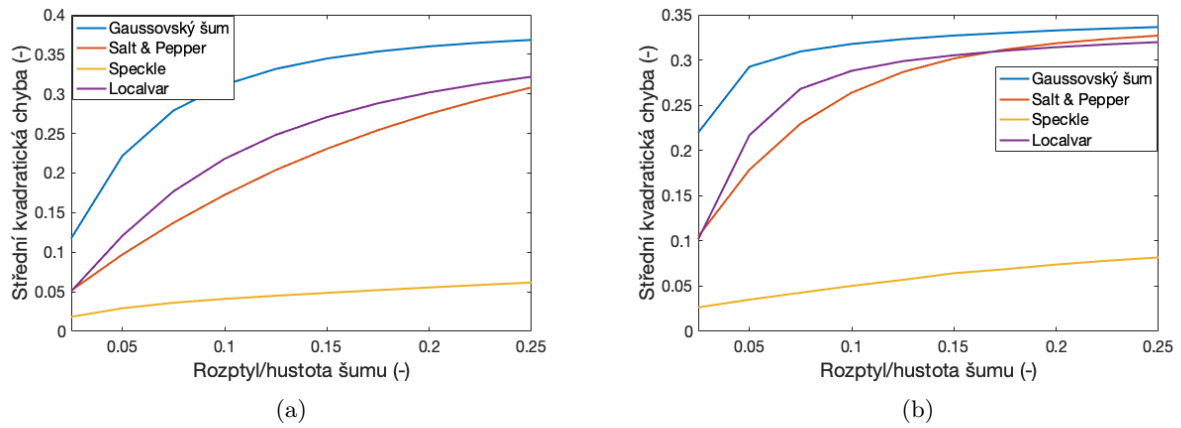
totožné s výsledků s vlastní implementací konvoluce, ovšem při použití vlastní implementace konvoluce jsou výpočetní časy několikanásobně vyšší.



Obrázek 36: Porovnání vývoje korelace po detekci hran s použitím nativní funkce MATLABu a vlastní implmelentací funkce a to jak s použitím nativní funkce pro konvoluci ,tak s vlastní implementací konvoluce pro metodu Sobel použitou na obrazy zašumělé a) Gaussovským šumem, b) šumem Salt & Pepper a pro metodu Roberts na obrazy zašumělé c)Gaussovským šumem, d) šumem Salt & Pepper

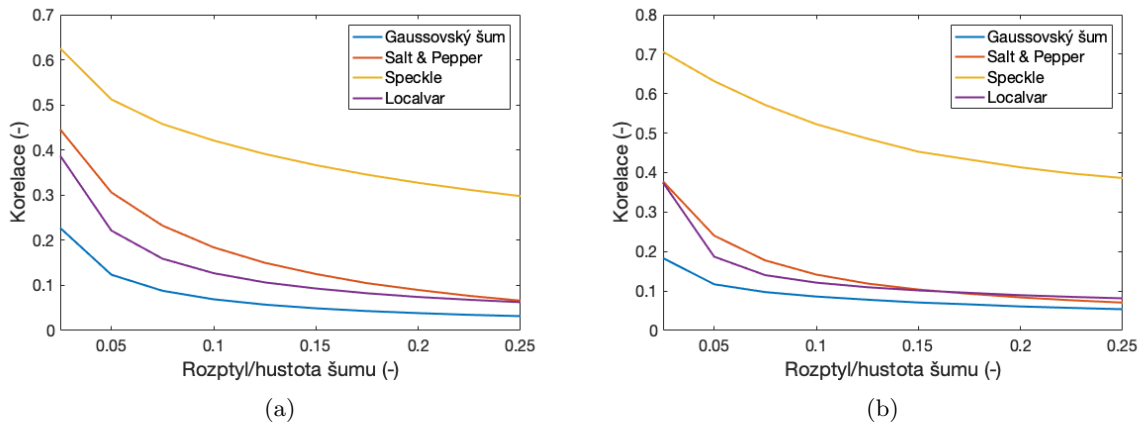
9.3 Vliv jednotlivých šumů na metody detekce hran

Tyto experimenty znázorňují, vliv jednotlivých šumů na dané metody pro detekci hran. Z grafů na obrázku 37 vyplývá, že největší vliv jak na metodu Sobel, tak na metodu Canny má Gaussovský šum, naopak nejméně je ovlivňuje šum Speckle.



Obrázek 37: Změna střední kvadratické chyby při změně šumu pro metodu a) Sobel, b) Canny

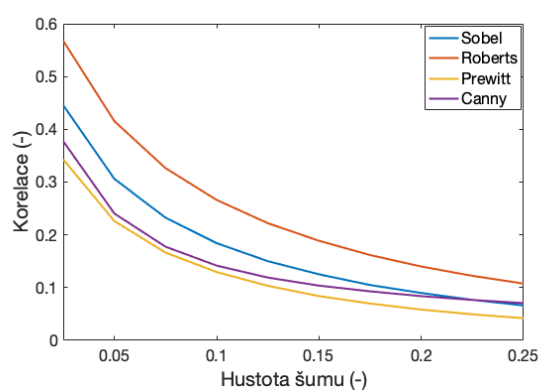
Následující grafy pozorují vliv jednotlivých šumů na změnu korelace pro jednotlivé metody. Výsledky jsou stejné jako u střední kvadratické chyby, tedy že největší vliv na detekci hran má Gaussovský šum a nejmenší šum Speckle.



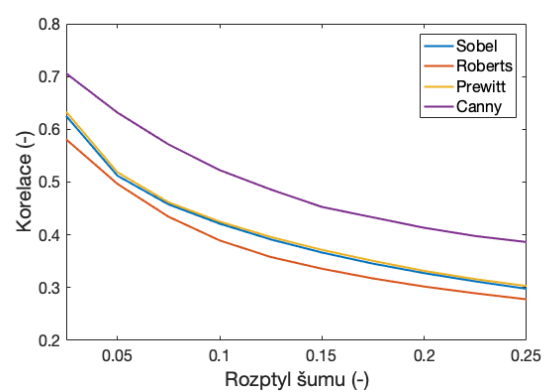
Obrázek 38: Změna korelace při změně šumu pro metodu a) Sobel, b) Canny

9.4 Porovnání jednotlivých metod

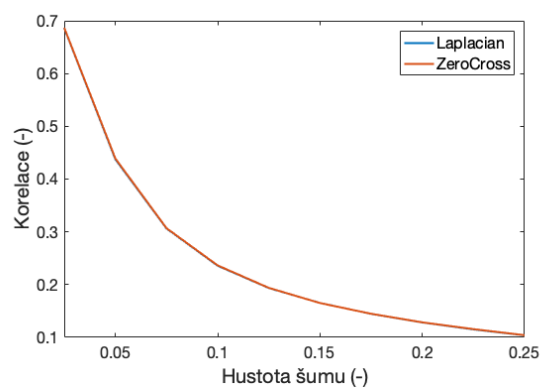
Tyto experimenty porovnávají jednotlivé metody mezi sebou a určují hodnotu korelace pro stejné parametry daných metod použitých na obrazech zašumělých šumy Salt & Pepper. Vývoj korelace můžeme vidět na obrázku 39. Z těchto grafů můžeme vyčíst, že pro obrazy zašumělé šumem Salt & Pepper vychází jako nejlepší metoda Roberts a pro obrazy zašumělé šumem Speckle metoda Canny. Metody Laplacian a ZeroCross podávají stejné výsledky, a to z důvodu použití Laplaceova filtru u metody ZeroCross.



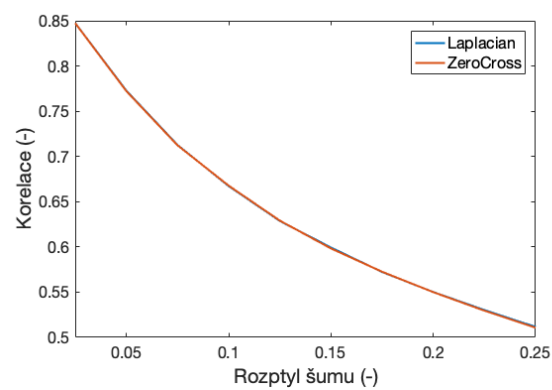
(a)



(b)

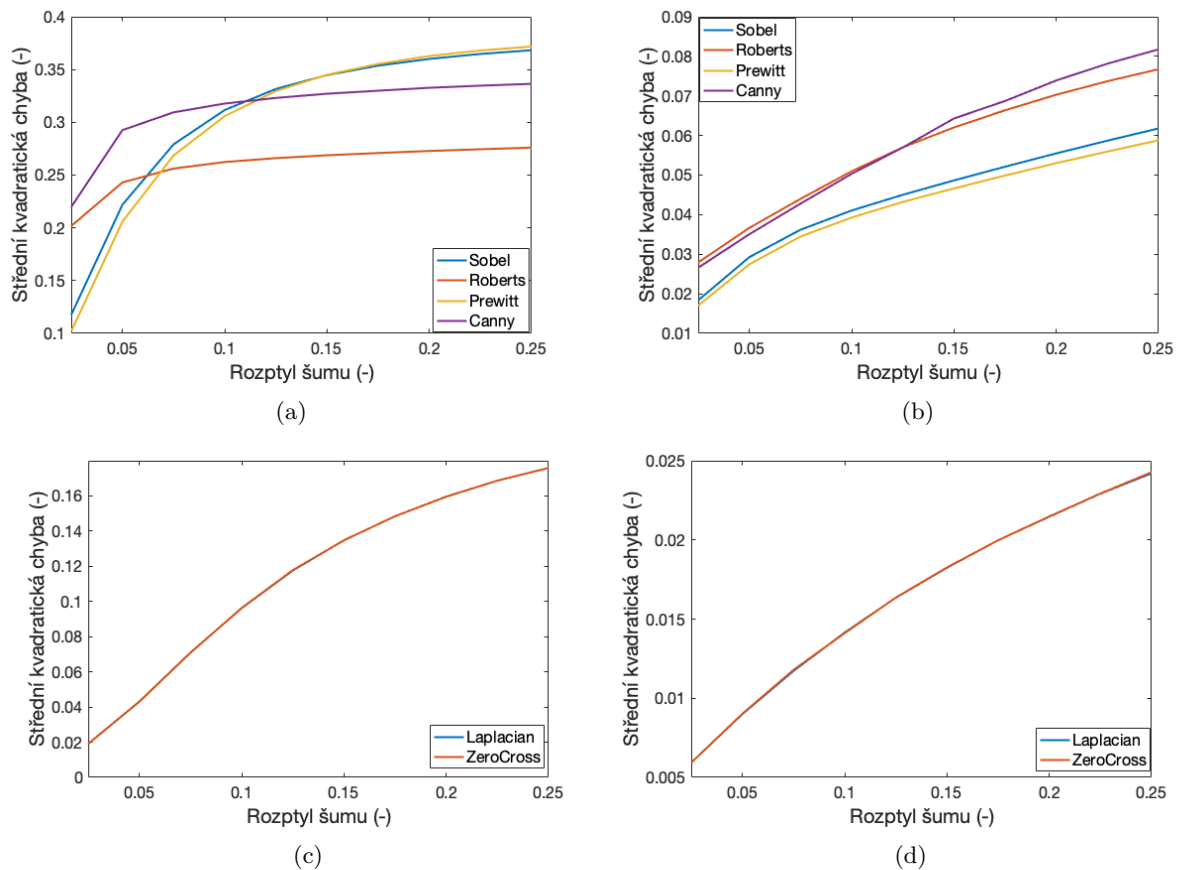


(c)



(d)

Obrázek 39: Porovnání výsledků korelace pro jednotlivé metody pro a), c) šum Salt&Pepper a b), d) šum Speckle s práhy a),b) 0.1, c),d) 0.01



Obrázek 40: Porovnání výsledků střední kvadratické chyby pro jednotlivé metody pro a), c) Gaussovský šum a b), d) šum Speckle s práhy a),b) 0.1, c),d) 0.01

9.5 Porovnání průměrných výsledků jednotlivých metod

Pro určení nejlepšího operátoru byly výsledky pro jednotlivé metody zprůměrovány napříč intervaly práhů a parametrů všech použitých šumových generátorů. Výsledky korelace znázorňuje následující tabulka.

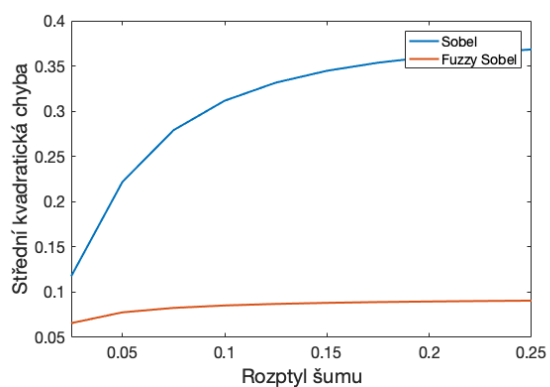
Tabulka 2: Tabulka průměrných hodnot korelace

Operátor	CT cévy	CT játra	MRI chrupavka	MRI cévy	Celkový průměr
Sobel	0.2184	0.1774	0.1133	0.2085	0.1794
Roberts	0.2205	0.1484	0.1020	0.2121	0.1707
Prewitt	0.2128	0.1737	0.1100	0.2042	0.1752
Canny	0.2401	0.2268	0.1704	0.2288	0.2165
Laplacián	0.3962	0.3088	0.2436	0.4028	0.3379
Zero Cross	0.3962	0.3086	0.2437	0.4028	0.3378

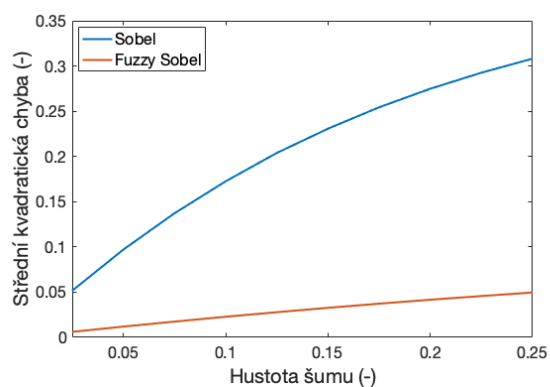
Absolutně nejlepších výsledků dosáhla metoda Zero Cross a Laplacián, tyto metody však využívají jiný interval prahů. Proto vyhodnocujeme nejlepší metodu využívající aproximaci první derivace, pro jež byl použit stejný interval prahů. Z těchto metod na základě průměrné korelace podává nejlepší výsledky metoda Canny.

9.6 Porovnání výsledků klasické metody Sobel a metody Sobel s použitím fuzzy logiky

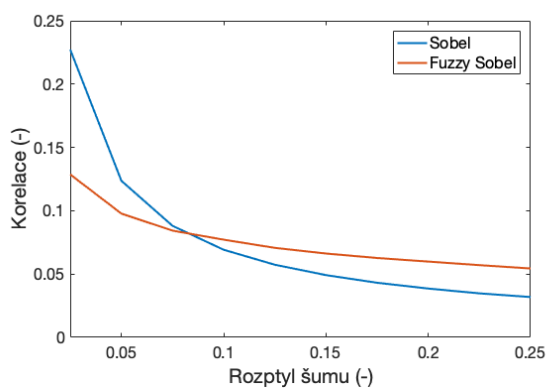
Poslední experimenty se zabývají porovnáním klasických metod pro detekci hran a metod pro detekci hran s použitím fuzzy logiky. Na obrázku 60 vidíme výsledky střední kvadratické chyby a korelace pro metodu Sobel a tutéž metodu s použitím fuzzy logiky. Výsledky střední kvadratické chyby (a), b)) ukazují, že metoda s použitím fuzzy logiky podává výrazně lepší výsledky než klasická metoda Sobel a to jak na obrazu zašuměném Gaussovským šumem a), tak se šumem Salt & Pepper b). Graf korelace na obrazech zašuměných šumem Salt & Pepper (d)) taktéž ukazuje, že metoda s použitím fuzzy logiky poskytuje lepší výsledky, ovšem na obrazech s Gaussovským šumem (c)) s nízkým rozplem se klasická metoda Sobel ukazuje jako výhodnější, při vyšším rozpletu však hodota korelace pro klasickou metodu Sobel výrazně klesá a opět můžeme říci, že metoda s použitím fuzzy logiky je vůči šumu odolnější.



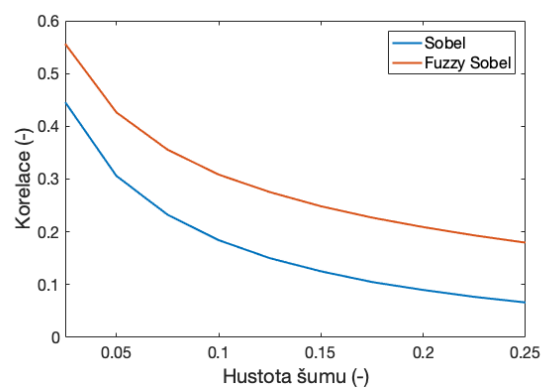
(a)



(b)



(c)

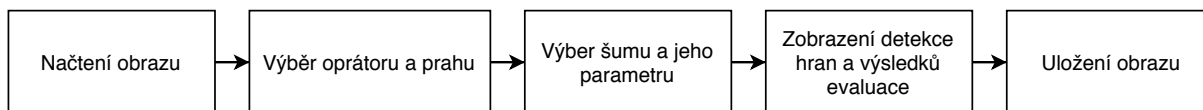


(d)

Obrázek 41: Porovnání výsledků střední kvadratické chyby a), b) a korelace c), d) pro metody Sobel a Sobel s použitím fuzzy logiky pro a), c) Gaussovský šum a b), d) šum Salt&Pepper.

10 Graficko-uživatelské rozhraní

Pro edukativní účely byla vytvořena aplikace, která jednoduše zobrazuje výsledky detekce hran na zašumělých i nezašumělých snímcích a zobrazuje jejich podobnost pomocí střední kvadratické chyby a konvoluce. Aplikace byla vytvořena v prostředí MATLAB pomocí nástroje App Designer.



Obrázek 42: Vývojový diagram pro práci s GUI

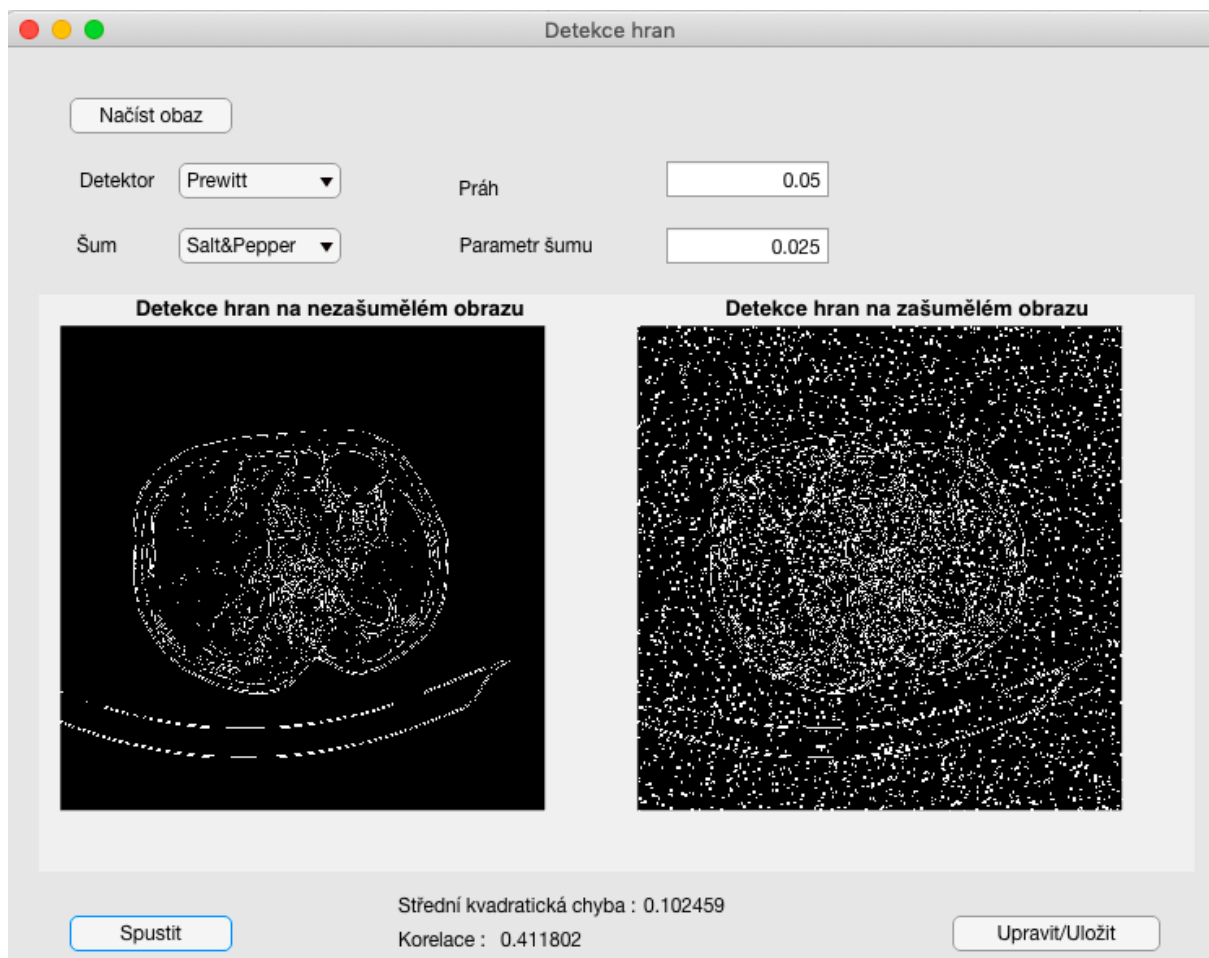
Po otevření aplikace se zobrazí prázdný formulář, kde je pomocí kliknutí na tlačítko *Načístobraz* vybrán jeden z obrázků dostupných datových sad.



Obrázek 43: Vstupní okno uživatelského prostředí

Obrázek se načte do levého okna pod nadpis *Originálníobraz*. Dalším krokem je volba metody pro detekci hran a šumového generátoru. Prostředí obsahuje všechny testované opeřátory a

šumy. Po zvolení operátoru a šumu určíme vhodný práh a parametr šumu. Po stisknutí tlačítka *Spustit* se spustí detekce hran, do levého okna se zobrazí původní nezašumělý obraz s hranami a v pravém okně dostaneme obraz s detekovanými hranami z původně zašumělého obrazu pomocí zvolených parametrů. Zároveň se ve stejný okamžik zobrazí výsledky evaluace těchto dvou obrazů a to konkrétně střední kvadratickou chybu a korelaci, jejichž výsledky se zobrazí pod výslednými obrazy.



Obrázek 44: Okno detekce hran

Po stisknutí tlačítka *Upravit/Uložit* se zobrazí obraz ze zašumělého snímku s detekovanými hranami v okně figure, které umožňuje přiblížení obrazu, zobrazení barevné mapy, přidání popisů a další. V neposlední řadě také umožňuje uložení obrazu v uživatelem zvoleném formátu.

11 Závěr

Tato práce popisuje základní úvod do zpracování obrazů a důležité pojmy v této tématice. Dále popisuje základní principy segmenetace obrazu, mezi které patří také hlavní téma této práce, detekce hran. Vybrané detektory hran byly nastudovány a teoreticky popsány. Byla vytvořena databáze obrazových dat pořízených na MRI a CT, na kterých byly provedeny experimenty. Tyto experimenty zahrnují sledování změny chování jednotlivých metod při změně typu dat, citlivost jednotlivých metod na různé šумы, porovnání vlastní implementace operátorů pro detekci hran a nativní funkce pro detekci hran a také porovnání účinnosti jednotlivých metod na snímcích zašuměných jedním typem šumu. Hlavním cílem této práce byla implementace experimentů, které nám umožnily určit nejvhodnější metodu detekce hran pro obrazy zašumělé různými typy šumů. Všechny experimenty proběhly v prostředí MATLAB. Výsledky experimentů byly hodnoceny pomocí střední kvadratické chyby a korelace. Nakonec bylo vytvořeno grafické uživatelské rozhraní sloužící pro edukativní účely. Závěry vyplývající z provedených experimentů :

- Není možné jednoznačně určit nejlepší metodu. Metody podávají různé výsledky v závislosti na typu a množství šumu.
- Změna datasetu nemá výrazný vliv na výsledky detekce hran, změna se projeví pouze při zašumění šumem Speckle.
- Největší vliv na kvalitu detekce hran má Gaussovský šum, nejméně ji ovlivňuje šum Speckle.
- Pro detekci hran obrazu zašumělého Gaussovským šumem získáme lepší výsledky s vlastní implementací funkce pro detekci hran.
- Metody s použitím fuzzy logiky podávají výrazně lepší výsledky než konvenční metody pro detekci hran.

Jedním z potenciálních zlepšení nebo rozšíření do budoucna by byla implementace dalších pokročilých metod jako je například MMG(Multiple Morphological Gradient), metoda Otsu nebo vlnková transformace.

Experimenty by v budoucnu mohly být snadno aplikované na další datové sady.

Literatura

- [1] HLAVÁČ, Václav, Milan ŠONKA. Počítačové vidění. Praha: Grada 1992, 272 s. ISBN 80-854-2467-3.
- [2] SOJKA, Eduard. Digitální zpracování a analýza obrazů. Ostrava: VŠB-Technická univerzita, 2000. ISBN 80-7078-746-5.
- [3] HÁJOVSKÝ, Radovan, Radka PUSTKOVÁ a František KUTÁLEK. Zpracování obrazu v měřicí a řídicí technice: učební text : studijní materiály pro studijní obor Měřicí a řídicí technika, Elektronika Fakulty elektrotechniky a informatiky. Ostrava: Vysoká škola báňská - Technická univerzita, 2012. ISBN 978-80-248-2596-0.
- [4] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. Image processing, analysis, and
- [5] ZMEŠKAL, Oldřich, Ondřej SEDLÁK a Martin NEŽÁDAL. Metody obrazové analýzy dat [online]. Ústav fyzikální a spotřební chemie, Fakulta chemická, Vysoké učení technické, Purkyňova 118, 612 00 Brno, 2002.
- [6] ZANATY, E. (2016). Medical Image Segmentation Techniques: An Overview. International Journal of informatics and medical data processing (JIMDP). 1. 1.
- [7] NOUREDDINE, Rasha, Khalil TARHINI a Soha SALEH. Segmentation and extraction of brain injury lesions from MRI images: Matlab implementation. In: 2015 International Conference on Advances in Biomedical Engineering (ICABME) [online]. IEEE, 2015, 2015, s. 45-48 [cit. 2019-04-18]. DOI: 10.1109/ICABME.2015.7323247. ISBN 978-1-4673-6516-1. Dostupné z: <http://ieeexplore.ieee.org/document/7323247/>
- [8] GONZALES, Rafael a Richard WOODS. Digital Image Processing. third edition. Upper Saddle River, NJ.: Prentice Hall, 2008. 3. ISBN 9780131687288.
- [9] Zpracování obrazu - 2. JPEG, RAW, TIFF a další formáty [online]. [cit. 2019-04-18]. Dostupné z: <http://www.fotografvani.cz/fotopraxe/zakladni-postupy1/zpracovani-obrazu-2-jpeg-raw-tiff-a-dalsi-formaty-151836cz>
- [10] Fribert, M.: Základy zpracování obrazu. Univerzita Pardubice, 2006, iISBN 80-7194-901-9.
- [11] Halounová, L.: Zpracování obrazových dat. Česká technika - nakladatelství ČVUT, 2009, ISBN 978-80-01-04253-3.
- [12] Deep learning for complete beginners: Using convolutional nets to recognise images, Cambridge Coding Academy
- [13] About DICOM. DICOM Library [online]. [cit. 2018-12-22]. Dostupné z: <https://www.dicomlibrary.com/dicom/>

- [14] SURI, Jasjit S, David L WILSON a Swamy LAXMINARAYAN. Handbook of biomedical image analysis. New York: Kluwer Academic/Plenum Publishers, c2005. ISBN 0-306-48605-9.
- [15] HLAVÁČ, Václav. Hledání hran [online]. České vysoké učení technické v Praze, Český institut informatiky, robotiky a kybernetiky.
- [16] ŽÁRA, Jiří, Bedřich BENEŠ a Petr FELKEL. Moderní počítačová grafika. Vyd. 1. Praha: Computer Press, 1998, xvi, 448 s. ISBN 80-722-6049-9.
- [17] FIŘT, Jaroslav a Radek HOLOTA. Digitalizace a zpracování obrazu [online]. , 1-5 [cit. 2018-12-22]. Dostupné z: <http://home.zcu.cz/~holota5/publ/DigZprO.pdf>
- [18] Obrazové filtry [online]. [cit. 2018-12-22]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=18431
- [19] HOROVÁ, Ivana a Jiří ZELINKA. Numerické metody. 2., rozš. vyd. Brno: Masarykova univerzita v Brně, 2004. ISBN 8021033177.
- [20] HLAVÁČ, Václav a Miloš SEDLÁČEK. Zpracování signálů a obrazů. Vyd. 3. V Praze: České vysoké učení technické, 2009. ISBN 978-80-01-04442-1.
- [21] KAUR, Ravneet, Ravtej Singh SANDHU, Ayush GERA a Tarlochan KAUR. Edge detection in digital panoramic dental radiograph using improved morphological gradient and MATLAB. In: 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)[online]. IEEE, 2017, 2017, s. 793-797 [cit. 2018-06-23]. DOI: 10.1109/SmartTechCon.2017.8358481. ISBN 978-1-5386-0569-1. Dostupné z: <https://ieeexplore.ieee.org/document/8358481/>
- [22] TRA, Pham N. H., Nguyen T. HAI a Tran T MAI. Image segmentation for detection of benign and malignant tumors. In: 2016 International Conference on Biomedical Engineering (BME-HUST) [online]. IEEE, 2016, 2016, s. 51-54 [cit. 2018-06-23]. DOI: 10.1109/BME-HUST.2016.7782105. ISBN 978-1-5090-1097-4. Dostupné z: <http://ieeexplore.ieee.org/document/7782105/>
- [23] SURVEY ON VARIOUS NOISES AND TECHNIQUES FOR DENOISING THE COLOR IMAGE [www.ijaiem.org]. 2013 [cit. 2019-04-11]. ISSN 2319 - 4847. Dostupné z: <https://www.ijaiem.org/volume2issue11/IJAIEM-2013-11-24-070.pdf>
- [24] ANAS, Essa. Edge detection techniques using fuzzy logic. In: 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN) [online]. IEEE, 2016, 2016, s. 169-173 [cit. 2019-04-11]. DOI: 10.1109/SPIN.2016.7566682. ISBN 978-1-4673-9197-9. Dostupné z: <http://ieeexplore.ieee.org/document/7566682/>

- [25] MAINI, Raman a Dr. Himanshu AGGARWAL. Study and Comparison of Various Image Edge Detection Techniques [online]. , 1-12 [cit. 2018-06-19]. Dostupné z: <http://www.cscjournals.org/manuscript/Journals/IJIP/Volume3/Issue1/IJIP-15.pdf>

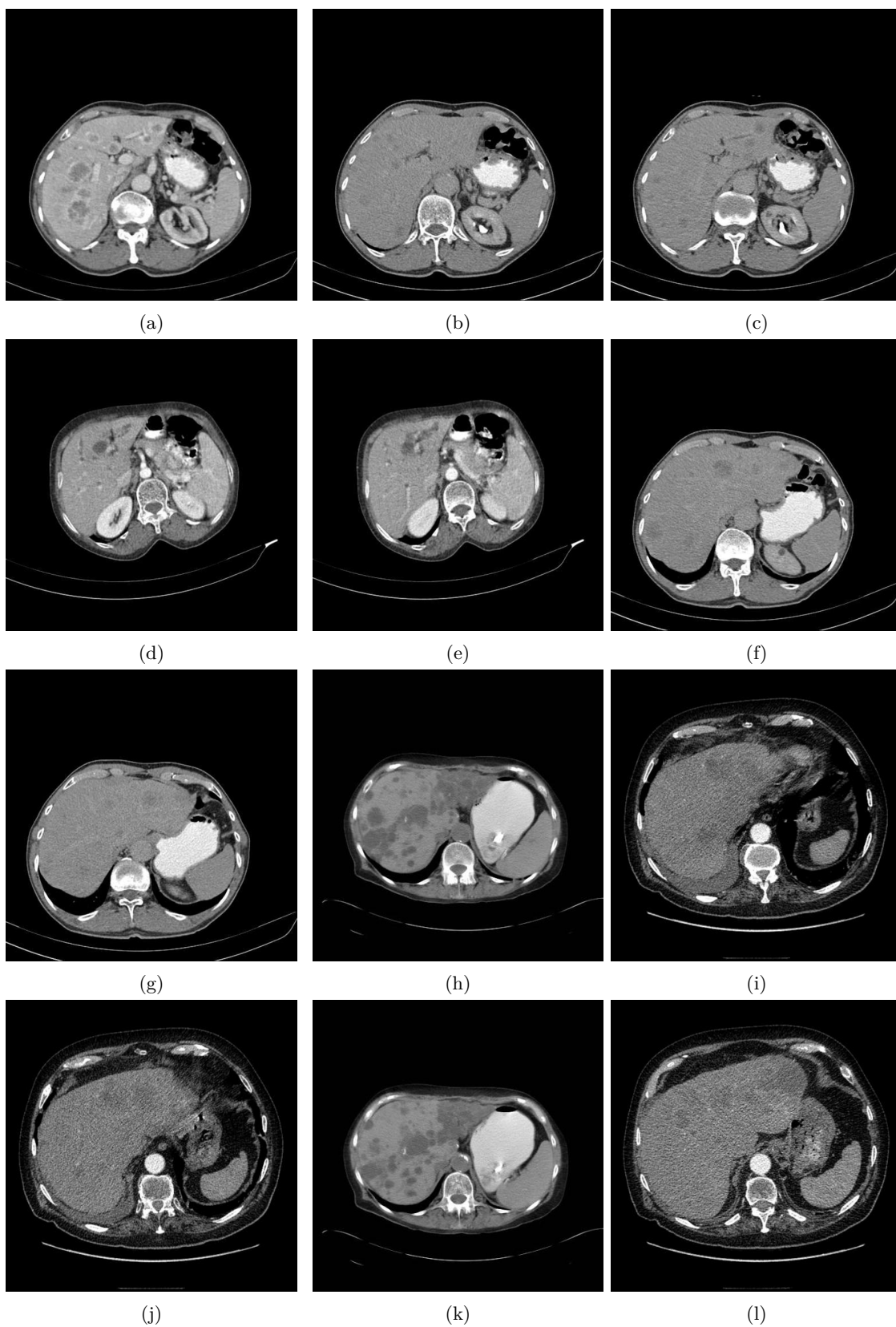
Seznam příloh

A Ukázka datasetů	67
B Ukázky zašumělých obrazů	71
B.1 Gaussovský šum	71
B.2 Šum Salt & Pepper	72
B.3 Šum Speckle	73
B.4 Šum Localvar	74
C Výstupy z GUI	75
D Porovnání metody Sobel s metodou Sobel s použitím fuzzy logiky	79
E Porovnání jednotlivých metod	80
F Vliv šumu na jednotlivé operátory	82
G Obsah CD	83

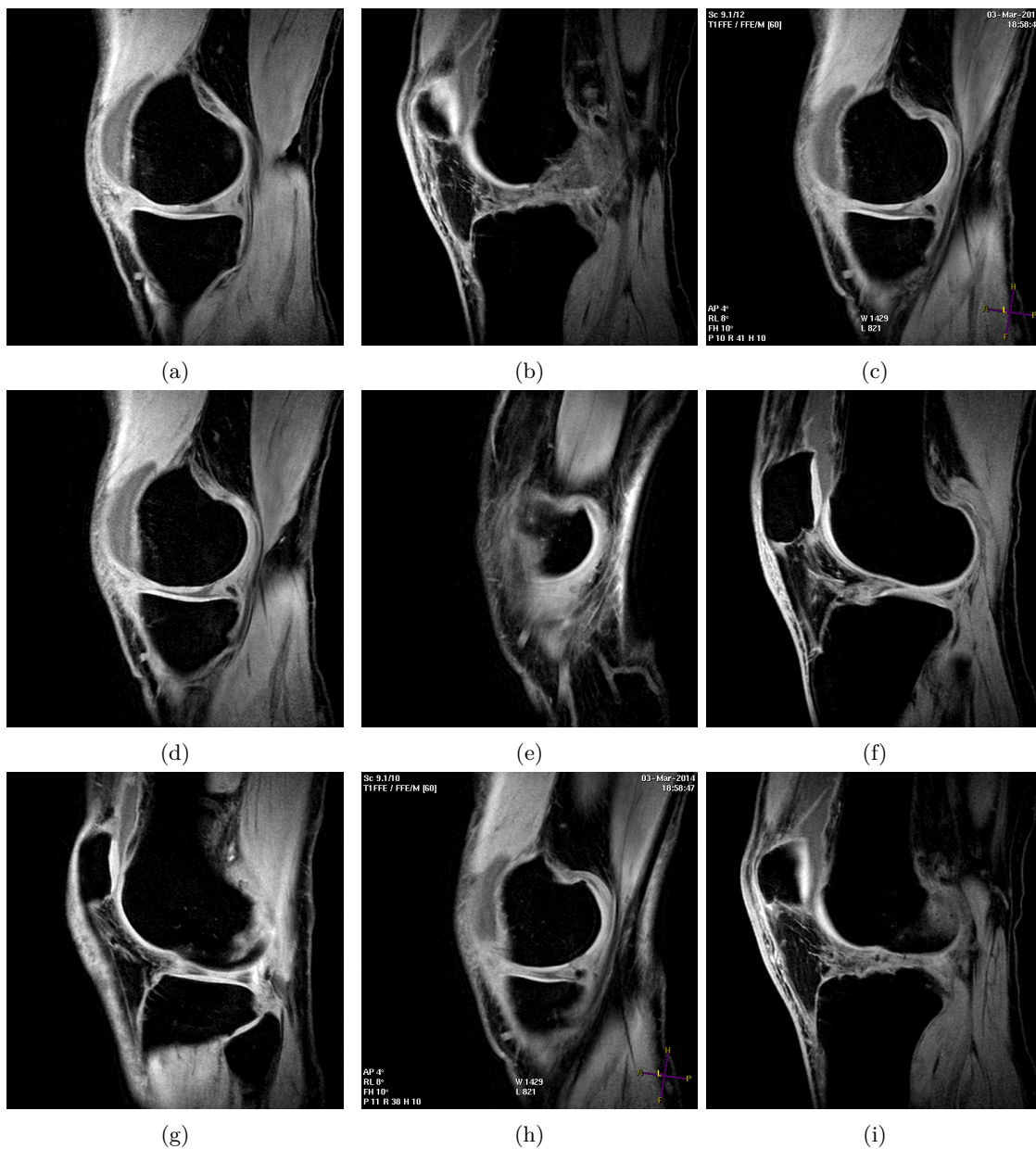
A Ukázka datasetů



Obrázek 45: Ukázka obrazů první datové sady CT obsahující kalcifikaci cév



Obrázek 46: Ukázka obrazů druhé datové sady CT jater



Obrázek 47: Ukázka obrazů třetí datové sady MRI kloubní chrupavky



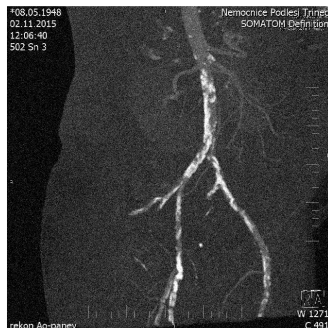
Obrázek 48: Ukázka obrazů čtvrté datové sady MRI krevních cév

B Ukázky zašumělých obrazů

B.1 Gaussovský šum



(a)



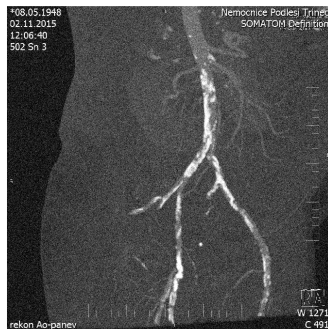
(b)



(c)



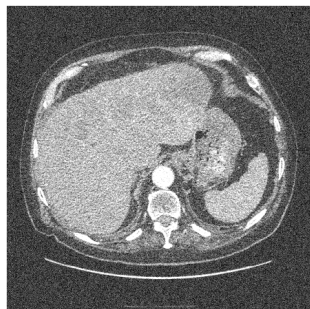
(d)



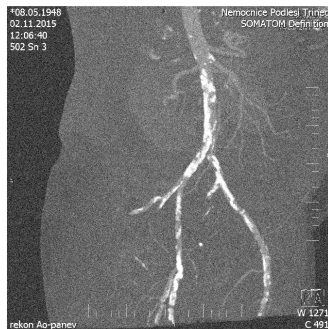
(e)



(f)



(g)



(h)



(i)

Obrázek 49: Gaussovský šum s rozptylem a)-c) 0.025, d)-f) 0.1, g)-i) 0.25

B.2 Šum Salt & Pepper



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

Obrázek 50: Šum Salt & Pepper s hustotou a)-c) 0.025, d)-f) 0.1, g)-i) 0.25

B.3 Šum Speckle



(a)



(b)



(c)



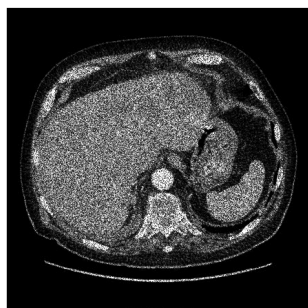
(d)



(e)



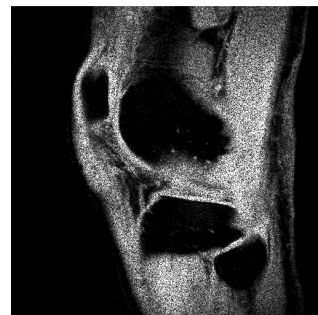
(f)



(g)



(h)



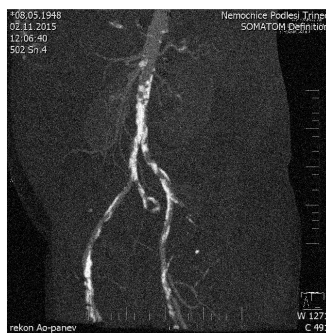
(i)

Obrázek 51: Šum Speckle s rozptylem a)-c) 0.025, d)-f) 0.1, g)-i) 0.25

B.4 Šum Localvar



(a)



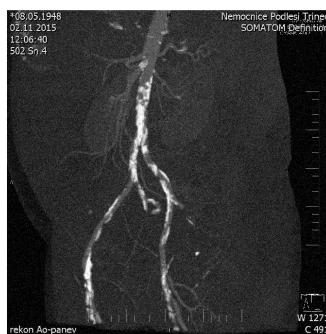
(b)



(c)



(d)



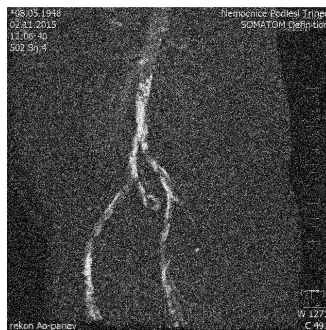
(e)



(f)



(g)



(h)



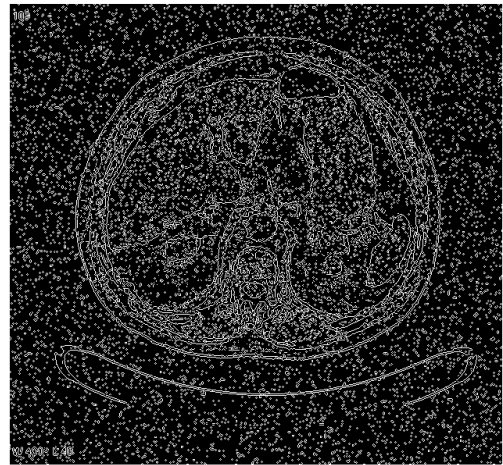
(i)

Obrázek 52: Šum Speckle s rozptylem a)-c) 0.025, d)-f) 0.1, g)-i) 0.25

C Výstupy z GUI

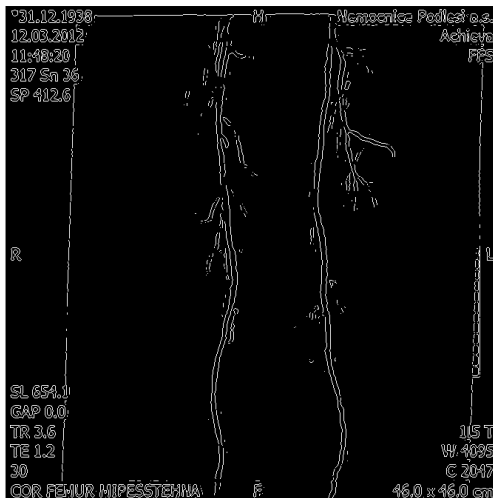


(a)

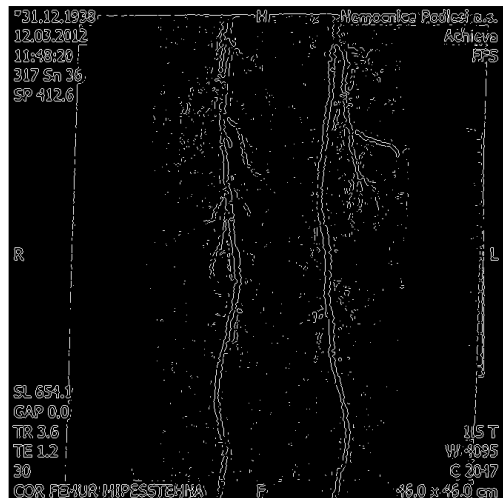


(b)

Obrázek 53: a) Sobel s práhem 0.05 b) Sobel s práhem 0.05 na obraze zašuměném šumem Salt & Pepper s hustotou 0.05



(a)

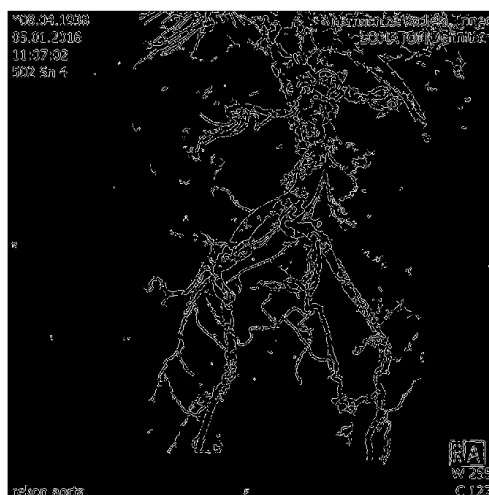


(b)

Obrázek 54: a) Laplacian s práhem 0.0075 b) Laplacian s práhem 0.0075 na obraze zašuměném šumem Speckle s rozptylem 0.1

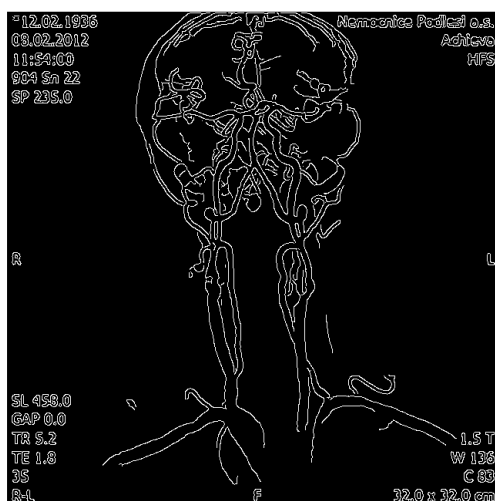


(a)

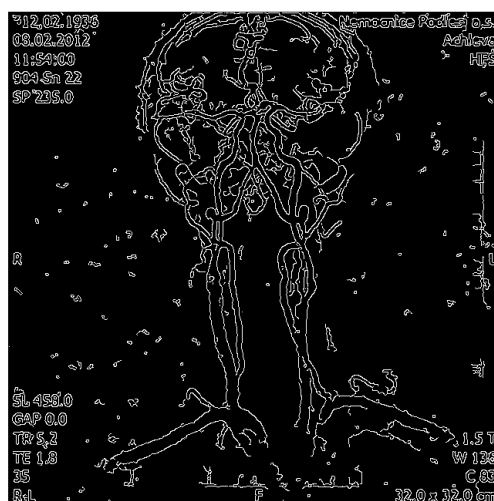


(b)

Obrázek 55: a) Canny s práhem 0.25 b) Canny s práhem 0.25 na obraze zašuměném šumem Localvar s rozptylem 0.05

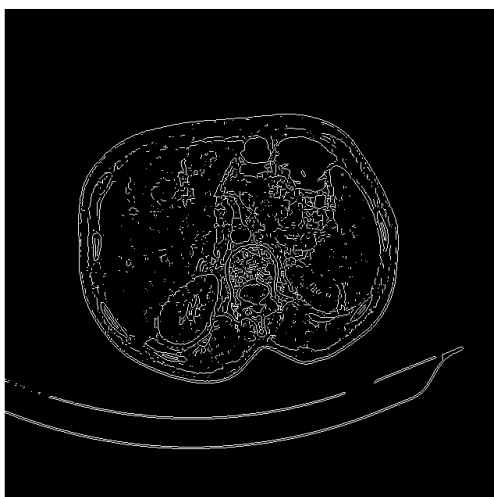


(a)

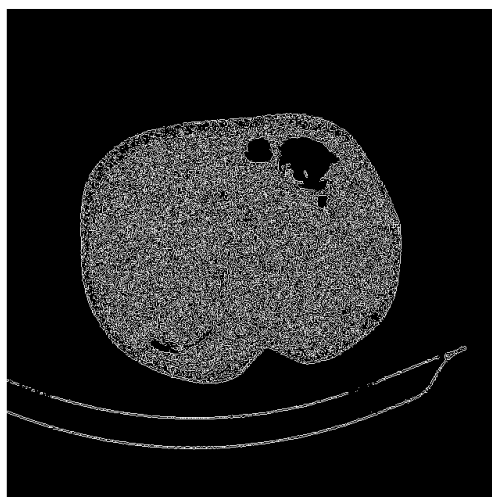


(b)

Obrázek 56: a) Canny s práhem 0.25 b) Canny s práhem 0.25 na obraze zašuměném šumem Salt & Pepper s hustotou 0.05

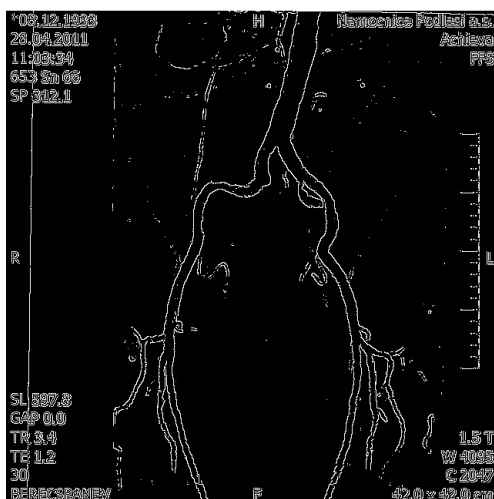


(a)

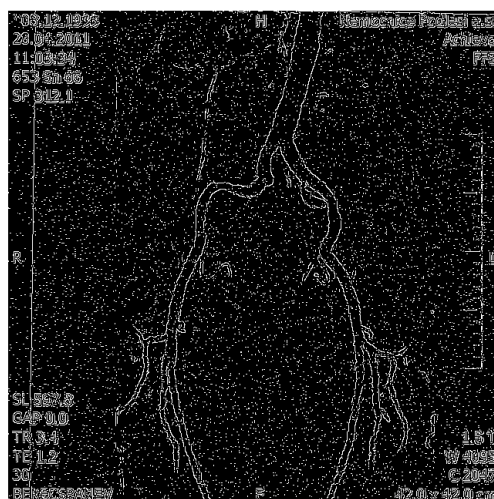


(b)

Obrázek 57: a) Prewitt s práhem 0.05 b) Prewitt s práhem 0.05 na obraze zašuměném šumem Speckle s rozptylem 0.25



(a)

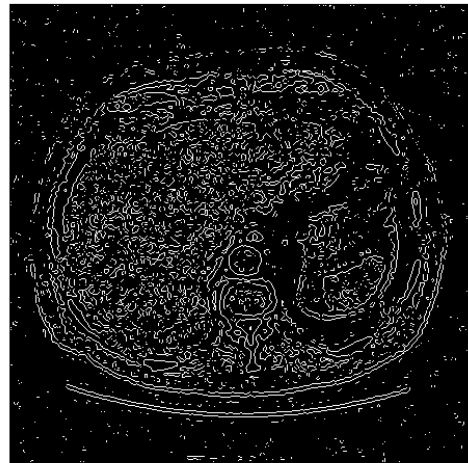


(b)

Obrázek 58: a) Roberts s práhem 0.05 b) Roberts s práhem 0.05 na obraze zašuměném šumem Salt & Pepper s hustotou 0.025



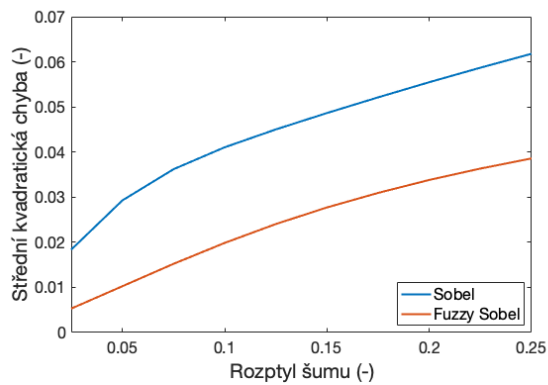
(a)



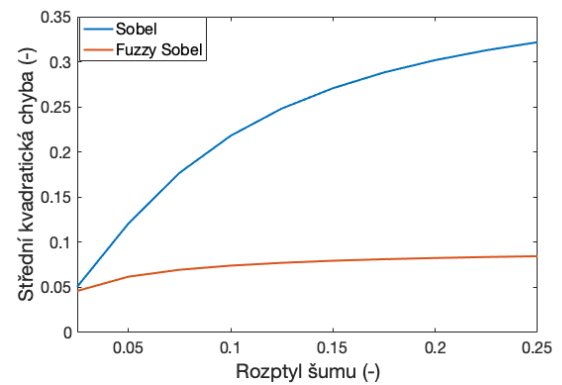
(b)

Obrázek 59: a) ZeroCross s práhem 0.0075 b) ZeroCross s práhem 0.05 na obraze zašuměném Gaussovským šumem s rozptylem 0.025

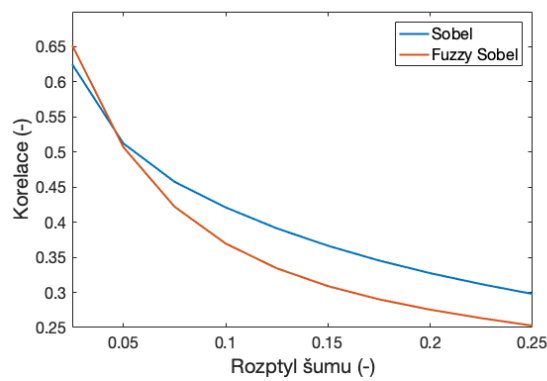
D Porovnání metody Sobel s metodou Sobel s použitím fuzzy logiky



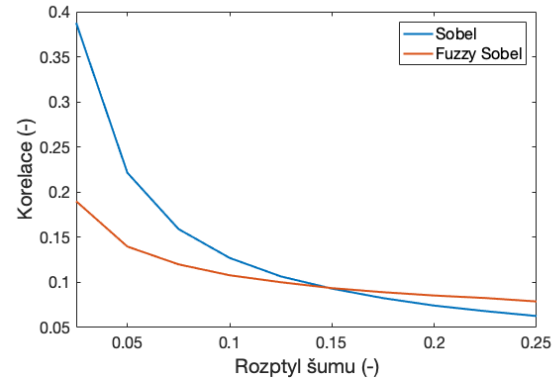
(a)



(b)



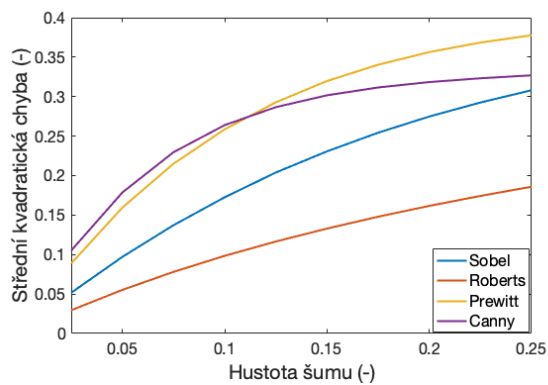
(c)



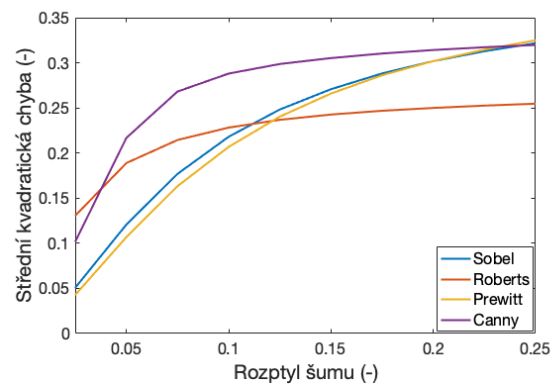
(d)

Obrázek 60: Porovnání výsledků střední kvadratické chyby a), b) a korelace c), d) pro metody Sobel a Sobel s použitím fuzzy logiky pro a), c) šum Speckle a b), d) šum Localvar.

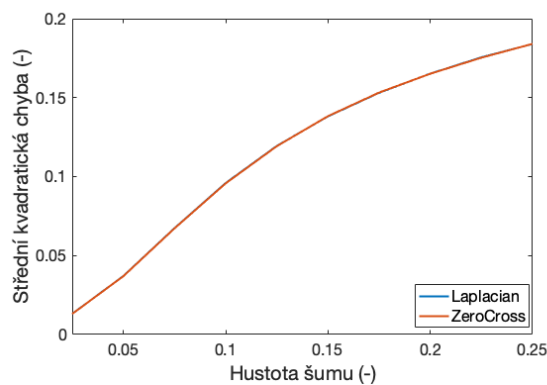
E Porovnání jednotlivých metod



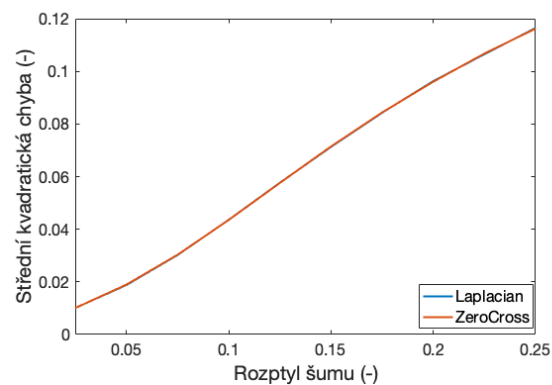
(a)



(b)

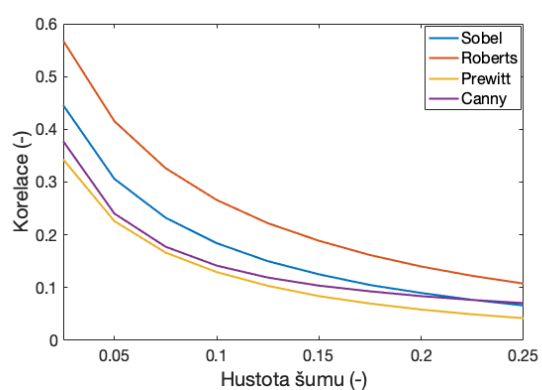


(c)

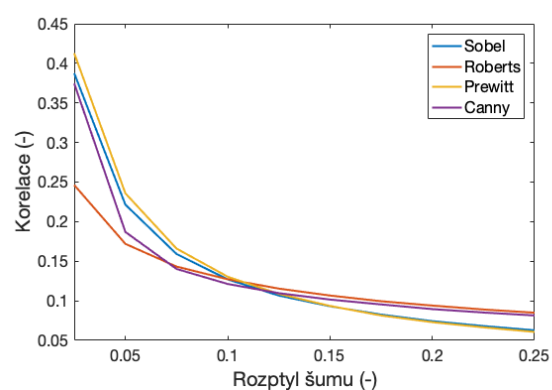


(d)

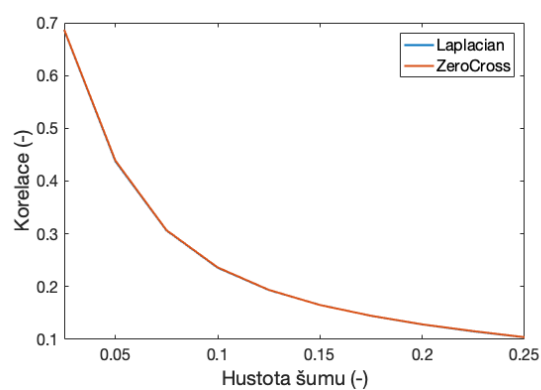
Obrázek 61: Porovnání výsledků střední kvadratické chyby pro jednotlivé metody pro a), c) Salt & Pepper a b), d) šum Localvar.



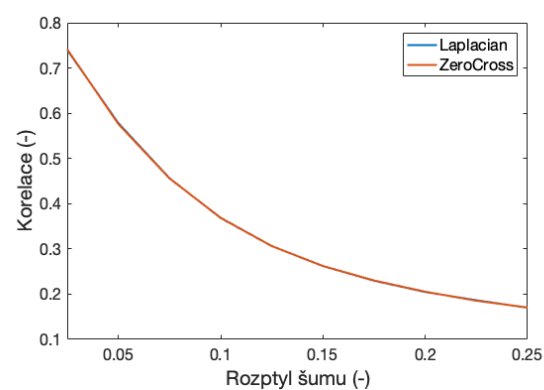
(a)



(b)



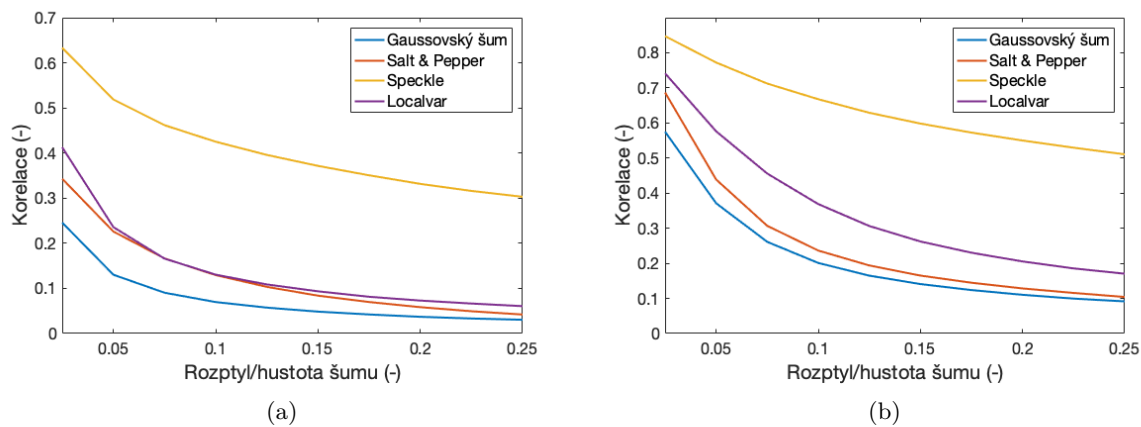
(c)



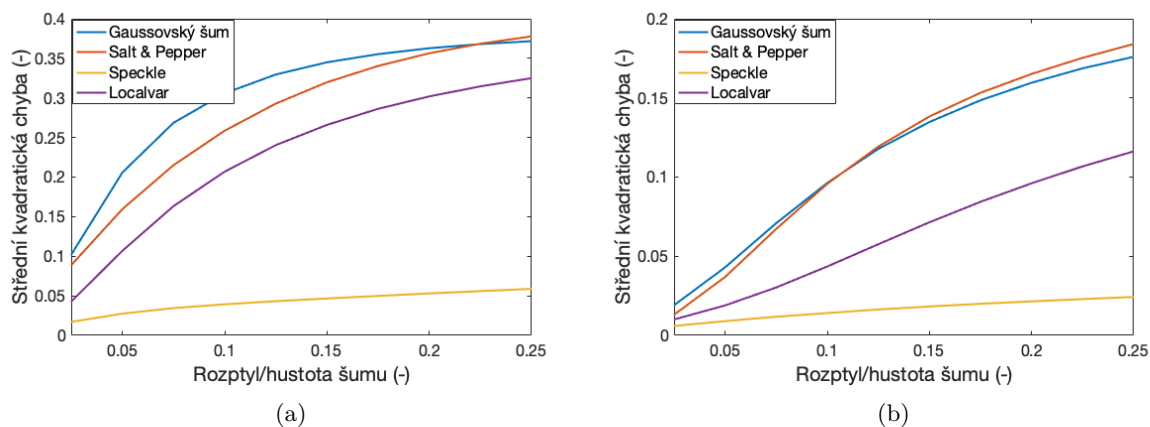
(d)

Obrázek 62: Porovnání výsledků korelace pro jednotlivé metody pro a), c) Salt & Pepper a b), d) šum Localvar.

F Vliv šumu na jednotlivé operátory



Obrázek 63: Změna korelace při změně šumu pro metodu a) Prewitt, b) ZeroCross



Obrázek 64: Změna střední kvadratické chyby při změně šumu pro metodu a) Prewitt, b) ZeroCross

G Obsah CD

- Bakalářská práce ve formátu PDF
- Zdrojové kódy
- Aplikace s uživatelským rozhraním
- Výsledky experimentů v souborech formátu *.mat